
UIResponderStandardEditActions Protocol Reference

Data Management: Event Handling



2010-02-25



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, 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

UIResponderStandardEditActions Protocol Reference 5

Overview 5

Tasks 5

 Handling Copy, Cut, Delete, and Paste Commands 5

 Handling Selection Commands 5

Instance Methods 6

 copy: 6

 cut: 6

 delete: 7

 paste: 7

 select: 8

 selectAll: 8

Document Revision History 11

UIResponderStandardEditActions Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/UIKit.framework
Declared in	UIResponder.h

Overview

The `UIResponderStandardEditActions` informal protocol declares methods that responder classes should override to handle common editing commands invoked in the user interface, such as Copy, Paste, and Select.

Although this is an informal protocol—that is, a category declared on `NSObject`—it is recommended that responder classes (that is, immediate or distant ancestors of `UIResponder`) override its methods. Starting with the first responder, `UIResponder` looks for a responder object that can handle the method, and works up the responder chain from there. Responder classes may also implement the `canPerformAction:withSender:` method of `UIResponder` to disable or enable user-interface commands based on the context. The [copy:](#) (page 6), [cut:](#) (page 6), [delete:](#) (page 7), [paste:](#) (page 7), [select:](#) (page 8), and [selectAll:](#) (page 8) methods are invoked when users tap the corresponding command in the menu managed by the `UIMenuController` shared instance.

Tasks

Handling Copy, Cut, Delete, and Paste Commands

- [copy:](#) (page 6) *required method*
Copy the selection to the pasteboard. (required)
- [cut:](#) (page 6) *required method*
Remove the selection from the user interface and write it to the pasteboard. (required)
- [delete:](#) (page 7) *required method*
Remove the selection from the user interface. (required)
- [paste:](#) (page 7) *required method*
Read data from the pasteboard and display it in the user interface. (required)

Handling Selection Commands

- [select:](#) (page 8) *required method*
Select the next object the user taps. (required)

- [selectAll:](#) (page 8) *required method*
Select all objects in the current view. (required)

Instance Methods

copy:

Copy the selection to the pasteboard. (required)

```
- (void)copy:(id)sender
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Copy command of the editing menu. A subclass of `UIResponder` typically implements this method. Using the methods of the `UIPasteboard` class, it should convert the selection into an appropriate object (if necessary) and write that object to a pasteboard. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

See Also

- [cut:](#) (page 6)
- [paste:](#) (page 7)

Availability

Available in iOS 3.0 and later.

Declared In

`UIResponder.h`

cut:

Remove the selection from the user interface and write it to the pasteboard. (required)

```
- (void)cut:(id)sender
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Cut command of the editing menu. A subclass of `UIResponder` typically implements this method. Using the methods of the `UIPasteboard` class, it should convert the selection into an appropriate object (if necessary) and write that object to a pasteboard. It should also remove the selected object from the user interface and, if applicable, from the application's data model. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

See Also

- [copy:](#) (page 6)
- [paste:](#) (page 7)

Availability

Available in iOS 3.0 and later.

Declared In

UIResponder.h

delete:

Remove the selection from the user interface. (required)

```
- (void)delete:(id)sender;
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Delete command of the editing menu. A subclass of `UIResponder` typically implements this method by removing the selected object from the user interface and, if applicable, from the application's data model. It should not write any data to the pasteboard. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

See Also

- [cut:](#) (page 6)

Availability

Available in iOS 3.2 and later.

Declared In

UIResponder.h

paste:

Read data from the pasteboard and display it in the user interface. (required)

```
- (void)paste:(id)sender
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Paste command of the editing menu. A subclass of `UIResponder` typically implements this method. Using the methods of the `UIPasteboard` class, it should read the data in the pasteboard, convert the data into an appropriate internal representation (if necessary),

and display it in the user interface. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

See Also

- [copy:](#) (page 6)
- [cut:](#) (page 6)

Availability

Available in iOS 3.0 and later.

Declared In

UIResponder.h

select:

Select the next object the user taps. (required)

```
- (void)select:(id)sender
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Select command of the editing menu. This command is used for targeted selection of items in the receiving view that can be broken up into chunks. This could be, for example, words in a text view. Another example might be a view that puts lists of visible objects in multiple groups; the `select:` command could be implemented to select all the items in the same group as the currently selected item.

A subclass of `UIResponder` typically implements this method. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

Availability

Available in iOS 3.0 and later.

See Also

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

Declared In

UIResponder.h

selectAll:

Select all objects in the current view. (required)

```
- (void)selectAll:(id)sender
```

Parameters

sender

The object calling this method.

Discussion

This method is invoked when the user taps the Select All command of the editing menu. A subclass of `UIResponder` typically implements this method by selecting all objects in the current view. The command travels from the first responder up the responder chain until it is handled; it is ignored if no responder handles it. If a responder doesn't handle the command in the current context, it should pass it to the next responder.

See Also

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

Availability

Available in iOS 3.0 and later.

Declared In

`UIResponder.h`

Document Revision History

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

Date	Notes
2010-02-25	Describes the delete: method added in iOS 3.2.
2009-05-19	Made minor corrections.
2009-03-08	First version of the document that describes the interface for handling editing commands.

REVISION HISTORY

Document Revision History