

---

# UIImagePickerControllerDelegate Protocol Reference

User Experience: Controls



2010-07-13



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

---

## **UIImagePickerControllerDelegate Protocol Reference 5**

---

Overview 5

Tasks 5

    Closing the Picker 5

Instance Methods 6

    UIImagePickerController:didFinishPickingMediaWithInfo: 6

    UIImagePickerControllerDidCancel: 6

Constants 7

    Editing Information Keys 7

## **Appendix A**

---

## **Deprecated UIImagePickerControllerDelegate Methods 9**

---

Deprecated in iOS 3.0 9

    UIImagePickerController:didFinishPickingImage:editingInfo: 9

---

## **Document Revision History 11**

---



# UIImagePickerControllerDelegate Protocol Reference

---

<b>Conforms to</b>	NSObject
<b>Framework</b>	/System/Library/Frameworks/UIKit.framework
<b>Availability</b>	Available in iOS 2.0 and later.
<b>Declared in</b>	UIImagePickerController.h

## Overview

The `UIImagePickerControllerDelegate` protocol defines methods that your delegate object must implement to interact with the image picker interface. The methods of this protocol notify your delegate when the user either picks an image or movie, or cancels the picker operation.

The delegate methods are responsible for dismissing the picker when the operation completes. To dismiss the picker, call the `dismissModalViewControllerAnimated:` method of the parent controller responsible for displaying the `UIImagePickerController` object.

To save a still image to the user's Saved Photos album, use the `UIImageWriteToSavedPhotosAlbum` function. To save a movie to the user's Saved Photos album, use the `UISaveVideoAtPathToSavedPhotosAlbum` function.

## Tasks

### Closing the Picker

- `UIImagePickerController:didFinishPickingMediaWithInfo:` (page 6)  
Tells the delegate that the user picked a still image or movie.
- `UIImagePickerControllerDidCancel:` (page 6)  
Tells the delegate that the user cancelled the pick operation.
- `UIImagePickerController:didFinishPickingImage:editingInfo:` (page 9) **Deprecated in iOS 3.0**  
Tells the delegate that the user picked an image. (**Deprecated.** Use `UIImagePickerController:didFinishPickingMediaWithInfo:` (page 6) instead.)

## Instance Methods

### **imagePickerController:didFinishPickingMediaWithInfo:**

Tells the delegate that the user picked a still image or movie.

```
- (void)imagePickerController:(UIImagePickerController *)picker
  didFinishPickingMediaWithInfo:(NSDictionary *)info
```

#### **Parameters**

*picker*

The controller object managing the image picker interface.

*info*

A dictionary containing the original image and the edited image, if an image was picked; or a filesystem URL for the movie, if a movie was picked. The dictionary also contains any relevant editing information. The keys for this dictionary are listed in [“Editing Information Keys”](#) (page 7).

#### **Discussion**

Your delegate object’s implementation of this method should pass the specified media on to any custom code that needs it, and should then dismiss the picker view.

When editing is enabled, the image picker view presents the user with a preview of the currently selected image or movie along with controls for modifying it. (This behavior is managed by the picker view prior to calling this method.) If the user modifies the image or movie, the editing information is available in the *info* parameter. The original image is also returned in the *info* parameter.

If you set the image picker’s `showsCameraControls` property to `NO` and provide your own custom controls, you can take multiple pictures before dismissing the image picker interface. However, if you set that property to `YES`, your delegate must dismiss the image picker interface after the user takes one picture or cancels the operation.

Implementation of this method is optional, but expected.

#### **Availability**

Available in iOS 3.0 and later.

#### **Declared In**

UIImagePickerController.h

### **imagePickerControllerDidCancel:**

Tells the delegate that the user cancelled the pick operation.

```
- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker
```

#### **Parameters**

*picker*

The controller object managing the image picker interface.

#### **Discussion**

Your delegate’s implementation of this method should dismiss the picker view by calling the `dismissModalViewControllerAnimated:` method of the parent view controller.

Implementation of this method is optional, but expected.

**Availability**

Available in iOS 2.0 and later.

**Declared In**

UIImagePickerController.h

## Constants

### Editing Information Keys

Keys for the editing information dictionary passed to the delegate.

```
NSString *const UIImagePickerControllerMediaType;
NSString *const UIImagePickerControllerOriginalImage;
NSString *const UIImagePickerControllerEditedImage;
NSString *const UIImagePickerControllerCropRect;
NSString *const UIImagePickerControllerMediaURL;
```

**Constants**

UIImagePickerControllerMediaType

Specifies the media type selected by the user.

The value for this key is an NSString object containing a type code such as kUTTypeImage or kUTTypeMovie.

Available in iOS 3.0 and later.

Declared in UIImagePickerController.h.

UIImagePickerControllerOriginalImage

Specifies the original, uncropped image selected by the user.

The value for this key is a UIImage object.

Available in iOS 2.0 and later.

Declared in UIImagePickerController.h.

UIImagePickerControllerEditedImage

Specifies an image edited by the user.

The value for this key is a UIImage object.

Available in iOS 3.0 and later.

Declared in UIImagePickerController.h.

UIImagePickerControllerCropRect

Specifies the cropping rectangle that was applied to the original image.

The value for this key is an NSValue object containing a CGRect opaque type.

Available in iOS 2.0 and later.

Declared in UIImagePickerController.h.

`UIImagePickerControllerMediaURL`

Specifies the filesystem URL for the movie.

The value for this key is an `NSURL` object.

Available in iOS 3.0 and later.

Declared in `UIImagePickerController.h`.



# Deprecated UIImagePickerControllerDelegate Methods

---

A method identified as deprecated has been superseded and may become unsupported in the future.

## Deprecated in iOS 3.0

### **imagePickerController:didFinishPickingImage:editingInfo:**

Tells the delegate that the user picked an image. (Deprecated in iOS 3.0. Use [imagePickerController:didFinishPickingMediaWithInfo:](#) (page 6) instead.)

```
- (void)imagePickerController:(UIImagePickerController *)picker
  didFinishPickingImage:(UIImage *)image
  editingInfo:(NSDictionary *)editingInfo
```

#### **Parameters**

*picker*

The controller object managing the image picker interface.

*image*

The image that the user picked. If user editing is enabled, this may be a cropped and adjusted version of the original image. In this case, the original image, and the editing information, are available in the *editingInfo* parameter.

*editingInfo*

A dictionary containing any relevant editing information. If editing is disabled, this parameter is `nil`. The keys for this dictionary are listed in [“Editing Information Keys”](#) (page 7).

#### **Discussion**

Your delegate’s implementation of this method should pass the specified image on to any custom code that needs it and then dismiss the picker view.

When user editing is enabled, the picker view presents the user with a preview of the currently selected image along with controls for modifying it. (This behavior is managed by the picker view prior to calling this method.) If the user modifies the image, the editing information is available in the *editingInfo* parameter. If you don’t need the editing information, simply use the image in the *image* parameter as is.

#### **Special Considerations**

This deprecated method supports picking only still pictures. The replacement method, [imagePickerController:didFinishPickingMediaWithInfo:](#) (page 6), supports picking movies as well as still pictures.

#### **Availability**

Available in iOS 2.0 and later.

Deprecated in iOS 3.0.

**Declared In**

UIImagePickerController.h

# Document Revision History

---

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

Date	Notes
2010-07-13	Minor changes.
2010-03-15	Updated for iOS 4.0.
	Updated the description of the <a href="#">UIImagePickerController:didFinishPickingMediaWithInfo:</a> (page 6) delegate method.
2009-07-07	Updated for iOS 3.1.
	Added information in the <a href="#">UIImagePickerControllerDelegate</a> (page 5) Overview section about related UIKit functions.
2009-06-15	Updated for iOS 3.0
	Added description of video recording support.
2008-10-15	Updated the guidance related to image sizes.
2008-05-27	New document that describes the protocol that defines methods for interacting with the image picker interface.

**REVISION HISTORY**

Document Revision History