
Image Views

[Cocoa](#) > [User Experience](#)



2002-11-12



Apple Inc.
© 2002 Apple Computer, 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 Cocoa 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

Introduction to Image Views 5

Organization of This Document 5

Using an Image View to Specify an Image 7

Setting an Image View's Appearance 9

Document Revision History 11

Introduction to Image Views

An image view displays a single image in a frame, and can optionally allow a user to drag an image to it.

Note that an image view performs its action only when a user drags an image to it. To perform an action when a user clicks an image, use a button For more information, see Buttons.

Organization of This Document

[“Setting an Image View’s Appearance”](#) (page 9) describes how to change the appearance of the image view’s image and frame. [“Using an Image View to Specify an Image”](#) (page 7) describes how to use an image view as an image well.

Using an Image View to Specify an Image

You can use an image view as an image well, which lets a user specify an image by dragging an image to it. Just use `setEditable:` with an argument of `YES`. When the user drags an image to the image view, the image view replaces its old image and sends its action message to its target.

If you don't want to use the image view as an image well, use `setEditable:` with an argument of `NO`.

Setting an Image View's Appearance

To set the image, use `setImage:`.

To set the frame style, use `setImageFrameStyle:` with one of these arguments:

- `NSImageFrameNone` shows an invisible frame
- `NSImageFramePhoto` shows a thin black outline and a dropped shadow
- `NSImageFrameGrayBezel` shows a gray, concave bezel that makes the image look sunken
- `NSImageFrameGroove` shows a thin groove that looks etched around the image
- `NSImageFrameButton` shows a convex bezel that makes the image stand out in relief, like a button

To choose how the image is scaled within its frame, use `setImageScaling:` with one of these arguments:

- `NSScaleProportionally`. If the image is too large, it shrinks to fit inside the frame. If the image is too small, it expands. The proportions of the image are preserved.
- `NSScaleToFit`. The image shrinks or expands, and its proportions distort, until it exactly fits the frame.
- `NSScaleNone`. The size and proportions of the image don't change. If the frame is too small to display the whole image, the edges of the image are trimmed off.

To choose how the image is aligned within its frame, use `setImageAlignment:` with one of these arguments:

- `NSImageAlignLeft`
- `NSImageAlignRight`
- `NSImageAlignCenter`
- `NSImageAlignTop`
- `NSImageAlignBottom`
- `NSImageAlignTopLeft`
- `NSImageAlignTopRight`
- `NSImageAlignBottomLeft`
- `NSImageAlignBottomRight`

Document Revision History

This table describes the changes to *Image Views*.

Date	Notes
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.

