

---

# NSImageCell Class Reference

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



2009-01-06



Apple Inc.  
© 2009 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, Cocoa, Mac, and Mac OS 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

---

## **NSImageCell Class Reference 5**

---

Overview	5
Adopted Protocols	5
Tasks	6
Aligning and Scaling the Image	6
Choosing the Frame	6
Instance Methods	6
imageAlignment	6
imageFrameStyle	7
imageScaling	7
setImageAlignment:	7
setImageFrameStyle:	8
setImageScaling:	8
Constants	9
NSImageAlignment	9
NSImageFrameStyle	10

---

## **Document Revision History 13**

---

## **Index 15**

---



# NSImageCell Class Reference

---

<b>Inherits from</b>	NSCell : NSObject
<b>Conforms to</b>	NSCoding NSCopying NSCoding (NSCell) NSCopying (NSCell) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Declared in</b>	NSImageCell.h
<b>Companion guides</b>	Image Views Matrix Programming Guide for Cocoa Table View Programming Guide
<b>Related sample code</b>	bMoviePalette bMoviePaletteCocoa Cropped Image

## Overview

An `NSImageCell` object displays a single image (encapsulated in an `NSImage` object) in a frame. This class provides methods for choosing the frame and for aligning and scaling the image to fit the frame.

The object value of an `NSImageCell` object must be an `NSImage` object, so if you use the `setObjectValue:` method of `NSCell`, be sure to supply an `NSImage` object as an argument. Because an `NSImage` object does not need to be converted for display, do not use the `NSCell` methods relating to formatters.

An `NSImageCell` object is usually associated with some kind of control object—an `NSImageView`, an `NSMatrix`, or an `NSTableView`.

## Adopted Protocols

`NSCoding`  
`encodeWithCoder:`  
`initWithCoder:`

## NSCopying

`copyWithZone:`

## Tasks

### Aligning and Scaling the Image

- [imageAlignment](#) (page 6)  
Returns the alignment of the receiver's image relative to its frame.
- [setImageAlignment:](#) (page 7)  
Sets the alignment of the image in its frame.
- [imageScaling](#) (page 7)  
Returns the scaling mode used to fit the receiver's image into the frame.
- [setImageScaling:](#) (page 8)  
Sets the scaling mode used to fit the receiver's image into the frame.

### Choosing the Frame

- [imageFrameStyle](#) (page 7)  
Returns the style of the frame that borders the image.
- [setImageFrameStyle:](#) (page 8)  
Sets the style of the frame that borders the image.

## Instance Methods

### imageAlignment

Returns the alignment of the receiver's image relative to its frame.

- (NSImageAlignment)imageAlignment

#### Return Value

One of the image alignment constants. For a list of possible values, see [NSImageAlignment](#) (page 9). The default value is `NSImageAlignCenter`.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [setImageAlignment:](#) (page 7)

#### Declared In

NSImageCell.h

## imageFrameStyle

Returns the style of the frame that borders the image.

- (NSImageFrameStyle)imageFrameStyle

### Return Value

One of the frame style constants. For a list of frame styles, see [NSImageFrameStyle](#) (page 10). The default value is `NSImageFrameNone`.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSImageCell.h

## imageScaling

Returns the scaling mode used to fit the receiver's image into the frame.

- (NSImageScaling)imageScaling

### Return Value

One of the image scaling constants. For a list of possible values, see `NSImageScaling`. The default value is `NSImageScaleProportionallyDown`.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSImageCell.h

## setImageAlignment:

Sets the alignment of the image in its frame.

- (void)setImageAlignment:(NSImageAlignment)alignment

### Parameters

*alignment*

One of the image alignment constants. For a list of possible values, see [NSImageAlignment](#) (page 9).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [imageAlignment](#) (page 6)

**Declared In**

NSImageCell.h

**setImageFrameStyle:**

Sets the style of the frame that borders the image.

```
- (void)setImageFrameStyle:(NSImageFrameStyle) frameStyle
```

**Parameters**

*frameStyle*

One of the frame style constants. For a list of frame styles, see [NSImageFrameStyle](#) (page 10).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [imageFrameStyle](#) (page 7)

**Related Sample Code**

bMoviePalette

bMoviePaletteCocoa

**Declared In**

NSImageCell.h

**setImageScaling:**

Sets the scaling mode used to fit the receiver's image into the frame.

```
- (void)setImageScaling:(NSImageScaling) scaling
```

**Parameters**

*scaling*

One of the image scaling constants. For a list of possible values, see [NSImageScaling](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [imageScaling](#) (page 7)

**Declared In**

NSImageCell.h



## Constants

### NSImageAlignment

These constants allow you to specify the location of the image in the frame and are used by [imageAlignment](#) (page 6) and [setImageAlignment:](#) (page 7).

```
typedef enum {
    NSImageAlignCenter = 0,
    NSImageAlignTop,
    NSImageAlignTopLeft,
    NSImageAlignTopRight,
    NSImageAlignLeft,
    NSImageAlignBottom,
    NSImageAlignBottomLeft,
    NSImageAlignBottomRight,
    NSImageAlignRight
} NSImageAlignment;
```

#### Constants

NSImageAlignCenter

**Center the image in the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

NSImageAlignTop

**Position the image along the top edge of the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

NSImageAlignTopLeft

**Align the image with the top and left edges of the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

NSImageAlignTopRight

**Align the image with the top and right edges of the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

NSImageAlignLeft

**Align the image with the left edge of the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

NSImageAlignBottom

**Align the image with the bottom edge of the cell.**

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

`NSImageAlignBottomLeft`

Align the image with the bottom and left edges of the cell.

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

`NSImageAlignBottomRight`

Align the image with the bottom and right edges of the cell.

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

`NSImageAlignRight`

Position the image along the right edge of the cell.

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSImageCell.h`

## NSImageFrameStyle

These constants allow you to specify the kind of frame bordering the image and are used by [imageFrameStyle](#) (page 7) and [setImageFrameStyle:](#) (page 8). These constants are obsolete, and are not compliant with the Apple Human Interface Guidelines:

```
typedef enum {
    NSImageFrameNone = 0,
    NSImageFramePhoto,
    NSImageFrameGrayBezel,
    NSImageFrameGroove,
    NSImageFrameButton
} NSImageFrameStyle;
```

#### Constants

`NSImageFrameNone`

An invisible frame

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

`NSImageFramePhoto`

A thin black outline and a dropped shadow

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

`NSImageFrameGrayBezel`

A gray, concave bezel that makes the image look sunken

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

### NSImageFrameGroove

A thin groove that looks etched around the image

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

### NSImageFrameButton

A convex bezel that makes the image stand out in relief, like a button

Available in Mac OS X v10.0 and later.

Declared in `NSImageCell.h`.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

`NSImageCell.h`



# Document Revision History

---

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

Date	Notes
2009-01-06	Added descriptions for the <code>NSImageAlignment</code> constants.
2008-10-15	Removed definition of the <code>NSImageScaling</code> constants, which are now documented in the <code>NSCell</code> class reference.
2007-01-22	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## I

---

`imageAlignment` **instance method** [6](#)  
`imageFrameStyle` **instance method** [7](#)  
`imageScaling` **instance method** [7](#)

## N

---

`NSImageAlignBottom` **constant** [9](#)  
`NSImageAlignBottomLeft` **constant** [10](#)  
`NSImageAlignBottomRight` **constant** [10](#)  
`NSImageAlignCenter` **constant** [9](#)  
`NSImageAlignLeft` **constant** [9](#)  
`NSImageAlignment` **data type** [9](#)  
`NSImageAlignRight` **constant** [10](#)  
`NSImageAlignTop` **constant** [9](#)  
`NSImageAlignTopLeft` **constant** [9](#)  
`NSImageAlignTopRight` **constant** [9](#)  
`NSImageFrameButton` **constant** [11](#)  
`NSImageFrameGrayBezel` **constant** [10](#)  
`NSImageFrameGroove` **constant** [11](#)  
`NSImageFrameNone` **constant** [10](#)  
`NSImageFramePhoto` **constant** [10](#)  
`NSImageFrameStyle` **data type** [10](#)

## S

---

`setImageAlignment:` **instance method** [7](#)  
`setImageFrameStyle:` **instance method** [8](#)  
`setImageScaling:` **instance method** [8](#)