

---

# **NSImageCell**

<b>Inherits From:</b>	NSCell : NSObject
<b>Conforms To:</b>	NSCopying, NSCoding (from NSCell) NSObject (from NSObject)
<b>Declared In:</b>	AppKit/NSImageCell.h

## Class Description

An NSImageCell displays a single NSImage 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 must be an NSImage, so if you use NSCell’s **setObjectValue:** method, be sure to supply an NSImage as an argument. Since an NSImage doesn’t need to be converted for display, you won’t use the NSCell methods relating to formatters.

An NSImageCell is usually associated with some kind of NSControl—an NSImageView, an NSMatrix, or an NSTableView. For more information, see the specifications for those classes.

## Method Types

Aligning and scaling the image	– imageAlignment – setImageAlignment: – imageScaling – setImageScaling:
Choosing the frame	– imageFrameStyle – setImageFrameStyle:

## Instance Methods

### **imageAlignment**

– (NSImageAlignment)**imageAlignment**

Returns the position of the cell’s image in the frame. For a list of possible alignments, see **setImageAlignment:.**

## **imageFrameStyle**

– (NSImageFrameStyle)**imageFrameStyle**

Returns the style of frame that appears around the image. For a list of frame styles, see **setImageFrameStyle:**.

## **imageScaling**

– (NSImageScaling)**imageScaling**

Returns the way that the cell’s image alters to fit the frame. For a list of possible values, see **setImageScaling:**.

## **setImageAlignment:**

– (void)**setImageAlignment:**(NSImageAlignment)*alignment*

Lets you specify the position of the image in the frame. The possible alignments are:

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

The default *alignment* is NSImageAlignCenter.

**See also:** – imageAlignment

## **setImageFrameStyle:**

– (void)**setImageFrameStyle:**(NSImageFrameStyle)*frameStyle*

Lets you specify the kind of frame that borders the image . The possible styles are:

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

---

The default *frameStyle* is `NSImageFrameNone`.

**See also:** – `imageFrameStyle`

## **setImageScaling:**

– (void)**setImageScaling:**(`NSImageScaling`)*scaling*

Lets you specify the way that the image alters to fit the frame. The possible values are:

- `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.

The default *scaling* is `NSScaleProportionally`.

The default *scaling* is `NSScaleProportionally`.

**See also:** – `imageScaling`