NSImageView

Inherits From:	NSControl: NSView: NSResponder: NSObject
Conforms To:	NSCoding (from NSResponder) NSObject (from NSObject)
Declared In:	AppKit/NSImageView.h

Class Description

An NSImageView displays a single NSImage in a frame. The NSImageView class provides methods for choosing the image, choosing the frame, and for aligning and scaling the image to fit the frame.

For an NSControl, NSImageView is quite limited in its ability to respond to user events: the only thing a user can do is drag in a new image. When it receives the new image, the NSImageView replaces its old image and sends its action message to its target. Even this low level of interactivity can be disabled: you can send the NSImageView the message **setEditable:**NO.

For more information, see the class specification for NSImageCell.

Method Types

Choosing the image	– image – setImage:
Choosing the frame	– imageFrameStyle – setImageFrameStyle:
Aligning and scaling the image	– imageAlignment – setImageAlignment: – imageScaling – setImageScaling:
Responding to user events	– isEditable – setEditable:

Instance Methods

image

– (NSImage *)**image**

Returns the NSImage displayed by the NSImageView.

See also: – **setImage:**

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:**.

isEditable

– (BOOL)**isEditable**

Returns whether the user can drag a new image into the frame. The default is YES.

See also: – **setEditable**:

setEditable:

– (void)**setEditable:**(BOOL)*flag*

Specifies whether the user can drag a new image into the frame.

See also: – **isEditable**

setImage:

– (void)**setImage:**(NSImage *)*image*

Lets you specify the image that the NSImageView displays.

See also: **– image**

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.

See also: – imageScaling