init
initFromImage:

Defining the cursor setImage:

image
setHotSpot:

Setting the cursor push

pop
+ pop
set
setOnMouseEntered:
setOnMouseExited:
mouseEntered:
mouseExited:

set, push, + pop, mouseEntered:, mouseExited:,

push

image

Returns the NXImage object that supplies the cursor image for the receiving NXCursor, or nil if no

initFromImage:, setImage:

init

Initializes the receiver, a newly allocated NXCursor instance, by sending it an initFromImage: mes
argument. This doesn't assign an image to the object. An image must then be set (with the setIma
cursor can be used. Returns self.

setImage:, initFromImage:

initFromImage:image

Initializes the receiver, a newly allocated NXCursor instance, by setting the image it will use to im
object. The image must be at least 16 pixels wide by 16 pixels high. If the image is smaller than 1
generated when the application tries to use the cursor, and the previous cursor remains in use. If th
16-by-16, only the lower-left 16-by-16 pixels of the image will be displayed. The default hot spot
corner of the 16-by-16 square.

This method is the designated initializer for the class. Returns self.

setHotSpot:, setImage:

mouseEntered:(NXEvent *)theEvent

Responds to a mouse-entered event by setting the receiver to be the current cursor, but only if enab
previous setOnMouseEntered: message. This method does not push the receiver on the cursor stac

setOnMouseEntered:

pop

Removes the topmost NXCursor object, not necessarily the receiver, from the cursor stack, and ma
down the current cursor. Returns self.

This method is a cover for the class method of the same name.

push

push

Puts the receiving NXCursor on the cursor stack and sets it to be the Window Server's cursor. Ret

This method can be used in conjunction with the pop method to manage a group of cursors within
push should be matched by a subsequent pop.

read:(NXTypedStream *)stream

Writes the NXCursor, including the image, to stream. Returns self.

write:

set

Makes the NXCursor the cursor displayed by the Window Server, and returns self. This method d
on the cursor stack.

setHotSpot:(const NXPoint *)aPoint

Sets the point on the cursor that will be used to report its location. The point is specified relative t
system with an origin at the upper left corner of the cursor image and coordinate units equal to tho
coordinate system. The point should not have any fractional coordinates, meaning that it should li
pixels. The point selects the pixel below it and to its right. This pixel is the one part of the cursor
never to be off-screen.

When the pixel selected by the hot spot lies inside a rectangle (say a button), the cursor is said to b
When the pixel is outside the rectangle, the cursor is taken to be outside the rectangle, even if othe
inside.

The default hot spot is at the upper left corner of the imageÐ (0,0) in its flipped coordinate system.

setImage:image

Assigns a new cursor image to the receiving NXCursor, and returns self. image should be an NX
image that's 16 pixels wide by 16 pixels high. If the image is smaller than 16-by-16, an error is ge
application tries to use the cursor, and the previous cursor remains in use. If the image is larger tha
lower-left 16-by-16 pixels of the image will be displayed.

Window's setTrackingRect:inside:owner:tag:left:right: method.

Cursor rectangles are a more convenient way of associating cursors with particular areas within a v

Returns self.

mouseEntered:, setTrackingRect:inside:owner:tag:left:right: (Window)

setOnMouseExited:(BOOL)flag

Determines whether the NXCursor should set itself to be the current cursor when it receives a mou
message. To be able to receive the event message, an NXCursor must first be made the owner of a
Window's setTrackingRect:inside:owner:tag:left:right: method.

Cursor rectangles are a more convenient way of associating cursors with particular areas within wi

Returns self.

mouseExited:, setTrackingRect:inside:owner:tag:left:right: (Window)

write:(NXTypedStream *)stream

Writes the NXCursor and its image to stream. Returns self.

read: