

`(id)setType:(int)aType` Sets how the NSButton highlights and shows its state.

`(id)setState:(int)value` Sets the NSButton's state to value (0 or 1).
`(id)state` Returns the NSButton's current state (0 or 1).

`(id)getPeriodicDelay:(float *)delay` Gets repeat parameters for continuous buttons.
`interval:(float *)interval`

`(id)setPeriodicDelay:(float)delay` Sets repeat parameters for continuous buttons.
`interval:(float)interval`

`(NSString *)alternateTitle` Returns the button's alternate title.

`(id)setAlternateTitle:(NSString *)aString` Makes aString the button's alternate title.

`(id)setTitle:(NSString *)aString` Makes aString the button's title.

`(NSString *)title` Returns the button's title.

`(NSImage *)alternateImage` Returns the button's alternate image.

`(NSImage *)image` Returns the button's image.

`(NSCellImagePosition)imagePosition` Returns the position of the button's image.

`(id)setAlternateImage:(NSImage *)anImage` Makes anImage the alternate image.

`(id)setImage:(NSImage *)anImage` Makes anImage the button's icon.

`(id)setImagePosition:(NSCellImagePosition)aPosition`

(id)highlight:(BOOL)flag

Highlights (or unhighlights) the button according to flag.

(NSString *)keyEquivalent

Returns the button's key equivalent.

(unsigned int)keyEquivalentModifierMask

Returns the mask indicating the possible modifier keys for keyEquivalent.

(id)setKeyEquivalent:(NSString *)aKeyEquivalent

Makes aKeyEquivalent the button's key equivalent.

(id)setKeyEquivalentModifierMask:(unsigned int)mask

Sets the mask that determines the possible modifier keys for keyEquivalent.

(id)performClick:(id)sender

Simulates the user's clicking the button.

(BOOL)performKeyEquivalent:(NSEvent *)anEvent

Simulates a mouse click, if the key in anEvent is right.