

Method	Description
<code>-(void)setType:(int)aType</code>	Sets how the NSButton highlights and shows its state.
<code>-(void)setState:(int)value</code>	Sets the NSButton's state to value (0 or 1).
<code>-(int)state</code>	Returns the NSButton's current state (0 or 1).
<code>-(void)getPeriodicDelay:(float *)delay interval:(float *)interval</code>	Gets repeat parameters for continuous buttons.
<code>-(void)setPeriodicDelay:(float)delay interval:(float)interval</code>	Sets repeat parameters for continuous buttons.
<code>-(NSString *)alternateTitle</code>	Returns the button's alternate title.
<code>-(void)setAlternateTitle:(NSString *)aString</code>	Makes aString the button's alternate title.
<code>-(void)setTitle:(NSString *)aString</code>	Makes aString the button's title.
<code>-(NSString *)title</code>	Returns the button's title.
<code>-(NSImage *)alternateImage</code>	Returns the button's alternate image.
<code>-(NSImage *)image</code>	Returns the button's image.
<code>-(NSCellImagePosition)imagePosition</code>	Returns the position of the button's image.
<code>-(void)setAlternateImage:(NSImage *)anImage</code>	Makes anImage the alternate image.
<code>-(void)setImage:(NSImage *)anImage</code>	Makes anImage the button's icon.
<code>-(void)setImagePosition:(NSCellImagePosition)aPosition</code>	

`(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 the key equivalent.

`(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 the key equivalent.

`(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.