

<code>NSString *)alternateTitle</code>	Returns the NSButtonCell's alternate title (used while the button is in the highlighted state).
<code>(id)setAlternateTitle:(NSString *)aString</code>	Makes a copy of aString and uses it as the NSButtonCell's alternate title.
<code>(id)setFont:(NSFont *)fontObject</code>	Sets the NSFont used to draw the title.
<code>(id)setTitle:(NSString *)aString</code>	Makes a copy of aString and uses it as the NSButtonCell's title.
<code>NSString *)title</code>	Returns the NSButtonCell's title.
<code>(NSImage *)alternateImage</code>	Returns the NSButtonCell's alternate image (used while the button is in the highlighted state).
<code>(NSCellImagePosition)imagePosition</code>	Returns the position of the NSButtonCell's image.
<code>(id)setAlternateImage:(NSImage *)anImage</code>	Makes anImage the alternate image.
<code>(id)setImagePosition:(NSCellImagePosition)aPosition</code>	Sets the position of the NSButtonCell's image in relation to its title.
<code>(id)getPeriodicDelay:(float *)delay interval:(float *)interval</code>	Gets repeat parameters for continuous NSButtonCells.
<code>(id)setPeriodicDelay:(float)delay interval:(float)interval</code>	Sets repeat parameters for continuous NSButtonCells.
<code>NSString *)keyEquivalent</code>	Returns the NSButtonCell's key equivalent.
<code>NSFont *)keyEquivalentFont</code>	Returns the NSFont used to draw the key equivalent.
<code>(signed int)keyEquivalentModifierMask</code>	Returns the mask indicating the possible modifier keys for NSButtonCell's key equivalent.
<code>(id)setKeyEquivalent:(NSString *)aKeyEquivalent</code>	Sets the NSButtonCell's key equivalent.

`(BOOL)isOpaque`

Returns whether receiver is opaque.

`(BOOL)isTransparent`

Returns whether the NSButtonCell is transparent.

`(void)setTransparent:(BOOL)flag`

Sets whether the NSButtonCell is transparent.

`(int)highlightsBy`

Returns how the NSButtonCell highlights when pressed.

`(void)setHighlightsBy:(int)aType`

Sets how the NSButtonCell highlights when pressed.

`(void)setShowsStateBy:(int)aType`

Sets how the NSButtonCell shows its alternate (pressed) state.

`(void)setType:(NSButtonType)aType`

Sets the NSButtonCell's display behavior.

`(int)showsStateBy`

Returns how NSButtonCell shows its alternate (pressed) state.

`(void)performClick:(id)sender`

Simulates a user's mouse click on the NSButtonCell.