MiscArrowButtonCell

Inherits From: NSButtonCell
Declared In: MiscAppKit/MiscArrowButtonCell.h

Class Description

With the help of MiscArrowButton, this is a recreation of the button (well, it was a View there) from Websters.app (in the Preferences panel). It's basically made to be used off of it's palette but you could always use it programmatically if you so choose.

There's just a few differences from using an NSButtonCell: I created a new designated initializer **initTextCell:alternateTitle:**, since it's convenient to be able to set both sides of the button at once. Also, since stringValue does not return anything useful (the string "0" or "1" depending upon it's current state), I made it return the the selected title instead.

Defined Types

SYNOPSIS

MiscArrowAlignment

```
typedef enum {
        MiscArrowAlignmentAbsolute = 0,
        MiscArrowAlignmentRelative = 1
} MiscArrowAlignment;
```

Instance Variables

float **cellHeight**;
unsigned int **arrowAlignment**;

cellHeight	We need to keep our cell height handy
arrowAlignment	Relative or absolute alignment?

Method Types

Class initialization	+ initialize
Initialization	- init - initTextCell: - initTextCell:alternateTitle:
Our state	- stringValue - arrowAlignment - setArrowAlignment:
Our size	- cellSizeForBounds: - titleRectForBounds: - alternateTitleRectForBounds:
Displaying	- drawInteriorWithFrame:inView: - drawWithFrame:inView:
Archiving	- initWithCoder: - encodeWithCoder:

Class Methods

initialize

+ (void)**initialize**

Sets our class version for archiving purposes.

Instance Methods

alternateTitleRectForBounds:

- (NSRect)**alternateTitleRectForBounds:**(NSRect)*theRect*

If the state is 1 (altContents selected), then this method will be called. It returns the location for editor to appear when editing the altContents. NOTE: This stopped working with OpenStep for Mach Prerelease 2. The button's IBEditors have changed.

arrowAlignment

- (MiscArrowAlignment)**arrowAlignment**

Returns MiscArrowAlignmentAbsolute or MiscArrowAlignmentRelative depending upon how alignment is currently being handled. See **setArrowAlignment:** for more details.

cellSizeForBounds:

- (NSSize)**cellSizeForBounds:**(NSRect)*theRect*

I believe this method is supposed to calculate the minimum size needed to fit the currently displayed contents (contents, altContents, and the arrow) in the cellframe.

drawInteriorWithFrame:inView:

- (void)**drawInteriorWithFrame:**(NSRect)*cellFrame*
  inView:(NSView*)*controlView*

This method is used to draw only the parts of the cell that do change when the state changes. Therefore, only the black arrow and the text are drawn here.

drawWithFrame:inView:

- (void)**drawWithFrame:**(NSRect)*cellFrame*
  inView:(NSView*)*controlView*

This part of the drawing displays only the parts that don't change often, which includes the diamond that encloses the arrow, and the border when I get around to adding one.

encodeWithCoder:

- (void)**encodeWithCoder:**(NSCoder*)*aCoder*

Encodes our instance variables.

init

- (id)**init**

Calls our designated initializer (**initTextCell:alternateTitle:**) with default titles ("Left" and "Right").

initTextCell:

- (id)**initTextCell:**(NSString*)*aString*

Calls our designated initializer. *aString* will become our left title. The right title will be set to "Right".

initTextCell:alternateTitle:

- (id)**initTextCell:**(NSString*)*aString*
  alternateTitle:(NSString*)*altString*

Our designated initializer. Sets both our title and alternate title.

initWithCoder:

- (id)**initWithCoder:**(NSCoder*)*aDecoder*

With luck this will still unarchive pre-OpenStep palettized code.

setArrowAlignment:

- (void)**setArrowAlignment:**(MiscArrowAlignment)*alignment*

Sets the *alignment* of the arrow. MiscArrowAlignmentAbsolute aligns the arrow in the center of the cellFrame, where MiscArrowAlignmentRelative centers the arrow between the left and right titles.

stringValue

- (NSString*)**stringValue**

Since **stringValue** in Button does not do much but return a string with a "0" or "1" depending upon our state, I overrode it to return the currently selected title.

titleRectForBounds:

- (NSRect)**titleRectForBounds:**(NSRect)*theRect*

As far as I can tell (which may not be all that far) is that this method is only used by the NSButton's IBEditor to tell how large the editor should be.