MiscCalendarView

Inherits From:	View : Object
Adopted Protocols:	TextDelegate, Drag and Drop
Declared In:	MiscKit/MiscCalendarView.h

Class Description

MiscCalendarView is a general calendar that supports the selection of days, months, and years. MiscCalendarView supports customizable header styles, colors, text delegation, and date delegation.

The palette view works fully under InterfaceBuilder. Its fully functional in test mode, and conforms to standard InterfaceBuilder usage. It also accepts drag-and-drop of color, just like textfield objects. Color can be dragged onto any of the displayed headers (month, year, inline month and year, or any of the day-of-the-week headers), and the text will change to that color. The day cells accept color as well. In the inspector panel you select which attribute you want to affect (cell background color, cell text color, cell background color when highlighted, cell text color when highlighted), and drag a color swatch onto any part of the matrix. You can also affect the color with the color well.

MiscCalendarView can use a custom date object for custom date behaviour. The date delegate can be set via IB, or programatically. MiscCalendarView uses a simple date object that works between the years of 1970 and 2037. Any custom date object must conform to the DateDelegate protocol, specified in the **Protocols** section.

MiscCalendarView can be set up as a text delegate. In turn, MiscCalendarView can be connected up to another object for text delegation. MiscCalendarView will pass through the text delegate messages, and on the textDidEnd:endChar: method, will update it's date.

MiscCalendarView can also be set up to accept drag-and-drop color when running in an application, so users can customize their colors if they wish.

External objects that respond to **setStringValue:** (such as textfields) can be attached to a MiscCalendarView. MiscCalendarView can display the month, year, and the full date in external objects. These connections can be made via IB, or programatically.

<<More could be said here, but I'm not sure what... "Dammit Jim, I'm a programmer, not a technical writer!">>

Instance Variables

id	**monthField;**
id	**yearField;**
id	**dateField;**
id	**dateDelegate;**
id	**textDelegate;**

monthField	An external object that responds to **setStringValue:** for displaying the month.
yearField	An external object that responds to **setStringValue:** for displaying the year.
dateField	An external object that responds to **setStringValue:** for displaying the date.
dateDelegate	An object that conforms to the **DateDelegate** protocol and provide date information.
textDelegate	An object that can provide text delegation to any text fields connected to the calendar via text delegation.

Method Types

Creating and freeing instances	±€initFrame:

Calendar properties	- setHighlightMode:
	- highlightMode
	- forHeader:display:
	- isHeaderDisplayed:
	- cellAt::lock:
	- forRow:lock:
	- forColumn:lock:
	- calendarMatrix
Working with color	- preserveCellColors:
	- cellColorsPreserved
	- setColorOf:to:forCellAt::
	- colorOf:forCellAt::
	- setColorOf:to:
	- colorOf:
	- setColorTo:forDay:
	- colorForDay:
	- allowColorDragAndDrop:
	- canDragAndDropColor
Working with dates	- setDateDelegate:
	- dateDelegate
	- displayDate:
	- updateDate:
	- incrementMonth:
	- decrementMonth:
	- incrementYear:
	- decrementYear:

Instance Methods

allowColorDragAndDrop:

± **allowColorDragAndDrop:**(BOOL)*yn*

Enables or disables the ability to drag and drop color into headers to set their color. Returns **self.**

See also: ± canDragAndDropColor

calendarMatrix

± **calendarMatrix**

Returns the underlying matrix used for the calendar.

canDragAndDropColor

± (BOOL)**canDragAndDropColor**

Returns YES if the headers can accept drag-and-drop color, otherwise returns NO**.**

See also: ± allowColorDragAndDrop:

cellAt::lock:

± **cellAt:**(int)*xPos* :(int)*yPos* **lock:**(BOOL)*yn*

Disables the cell at the specified *xPos* and *yPos* coordinates. Returns self.

See also: ± forRow:lock:, ± forColumn:lock:

cellColorsPreserved

± (BOOL)**cellColorsPreserved**

Returns YES or NO if individual cells that have had their colors changed (i.e. their background color) inherit global changes (i.e. the color of all cell backgrounds) set via a method like ± **setColorOf:to:**.

See also: ± preserveCellColors:

colorForDay:

± (NXColor)**colorForDay:**(MiscCVDays)*aDay*

Returns the color of the header for the specified day.

See also: ± setColorTo:forDay:

colorOf:forCellAt::

± (NXColor)**colorOf:**(int)*element* **forCellAt:**(int)*xPos* **:**(int)*yPos*

Return the color of the specified *element* for the cell at the matrix coordinates *xPos*,*yPos*. The argument *element* can be one of the following constants:

MISC_CV_CELLBACKGROUND
MISC_CV_CELLTEXT
MISC_CV_CELLHIGHLIGHT
MISC_CV_CELLTEXTHIGHLIGHT

See the section **Constants and Defined Types** for a description of the constants.

Returns NX_COLORBLACK if none of the elements above are specified.

See also: ± setColorOf:to:forCellAt::, ± setColorOf:to:, ± preserveCellColors:, ± colorOf:

colorOf:

± (NXColor)**colorOf:**(int)*element*

Returns the color of the specified *element.* In the cases where *element* specifies a cell color, the returned value is the default global color used. Individual cells may have different color values from the global ones. See **colorOf:forCellAt::** for information on how to get individual cell color attributes.

The argument *element* can be one of the following constants:

MISC_CV_CELLBACKGROUND
MISC_CV_CELLTEXT
MISC_CV_CELLHIGHLIGHT
MISC_CV_CELLTEXTHIGHLIGHT
MISC_CV_MONTHHEADER
MISC_CV_YEARHEADER

MISC_CV_INLINEMONTHHEADER
MISC_CV_INLINEYEARHEADER

See the section **Constants and Defined Types** for a description of the constants.

Returns NX_COLORBLACK if none of the elements above are specified.

See also: ± setColorOf:to:forCellAt::, ± setColorOf:to:, ± preserveCellColors:, ± colorOf:cellAt::

dateDelegate
± **dateDelegate**

Returns the date object used for day, date, and year calculations. Conforms to the **DateDelegate** protocol.

See also: ± setDateDelegate:, ± updateDate:, ± displayDate:

decrementMonth:
± **decrementMonth:***sender*

Sends the message **decrementMonth** to the date delegate and then calls **updateDate:** to display the new date. See the **DateDelegate** protocol for more information. Returns **self**.

See also: ± incrementMonth:,± setDateDelegate:, ± updateDate:, ± displayDate:

decrementYear:
± **decrementYear:***sender*

Sends the message **decrementYear** to the date delegate and then calls **updateDate:** to display the new date. See the **DateDelegate** protocol for more information. Returns **self**.

See also: ± incrementYear:,± setDateDelegate:, ± updateDate:, ± displayDate:

displayDate:

± **displayDate:***sender*

Causes the current date in the date object to be displayed. Returns **self**.

See also:± setDateDelegate:, ±dateDelegate, ± updateDate:

forColumn:lock:

± **forColumn:**(int)*col* **lock:**(BOOL)*yn*

Locks or unlocks the matrix column *col***,** based on the value of *yn*. Returns **self**.

See also: ± forRow:lock:, ± cellAt::lock:

forHeader:display:

± **forHeader:**(int)*aHeader* **display:**(BOOL)*doDisplay*

Turns on or off the displaying of the specified header, based on the value of *doDisplay*. Returns **self**.

The argument *aHeader* can be one of the following constants:

MISC_CV_DOWHEADER
MISC_CV_MONTHHEADER
MISC_CV_YEARHEADER
MISC_CV_MONTHANDYEARHEADER

See the section **Constants and Defined Types** for a description of the constants.

See also: ± isHeaderDisplayed:

forRow:lock:

± **forRow:**(int)*col* **lock:**(BOOL)*yn*

Locks or unlocks the matrix column *row***,** based on the value of *yn*. Returns **self**.

See also: ± forColumn:lock:, ± cellAt::lock:

highlightMode

± (int)**highlightMode**

Returns the highlight mode of the matrix. See documentation on **Matrix** for a description of what modes are available.

See also: ± setHighlightMode:, mode(Matrix)

incrementMonth:

± **incrementMonth:***sender*

Sends the message **incrementMonth** to the date delegate and then calls **updateDate:** to display the new date. See the **DateDelegate** protocol for more information. Returns **self**.

See also: ± decrementMonth:,± setDateDelegate:, ± updateDate:, ± displayDate:

incrementYear:

± **incrementYear:***sender*

Sends the message **incrementYear** to the date delegate and then calls **updateDate:** to display the new date. See the **DateDelegate** protocol for more information. Returns **self**.

See also: ± decrementYear:,± setDateDelegate:, ± updateDate:, ± displayDate:

initFrame:

± **initFrame:**(NXRect *)*aRect*

Initializes the view. This method is responsible for creating the underlying matrix, the header views, and the default date delegate object. Returns **self**.

isHeaderDisplayed:

± (BOOL)**isHeaderDisplayed:**(int)*aHeader*

Returns YES or NO, based on whether or not *aHeader* is being displayed.

The argument *aHeader* can be one of the following constants:

MISC_CV_DOWHEADER
MISC_CV_MONTHHEADER
MISC_CV_YEARHEADER
MISC_CV_MONTHANDYEARHEADER

See the section **Constants and Defined Types** for a description of the constants.

See also: ±forHeader:display:

preserveCellColors:

± **preserveCellColors:**(BOOL)*yn*

If *yn* is set to YES, then any cell that has had any of its color parameters (text color, background color, highlight color, or text highlight color) changed will not be affected by a global color change (see **setColorOf:**). If *yn* is set to NO, any global color change will affect all cells. Default behavior is NO. Returns **self**.

See also: ± cellColorsPreserved

setColorOf:to:

± **setColorOf:**(int)*element* **to:**(NXColor)*aColor*

Sets the color of the specified *element.* In the cases where *element* specifies a cell color, all matrix cells are affected, unless **preserveCellColors:** has been set to YES. Individual cells may have different color values from the global ones. See **setColorOf:to:forCell::** for information on how to set individual cell color attributes.

The argument *element* can be one of the following constants:

MISC_CV_CELLBACKGROUND
MISC_CV_CELLTEXT
MISC_CV_CELLHIGHLIGHT
MISC_CV_CELLTEXTHIGHLIGHT
MISC_CV_MONTHHEADER

MISC_CV_YEARHEADER
MISC_CV_INLINEMONTHHEADER
MISC_CV_INLINEYEARHEADER

See the section **Constants and Defined Types** for a description of the constants. Returns **self.**

See also: ± colorOf:forCellAt::, ± colorOf:, ± preserveCellColors:, ± setColorOf:to:forCellAt::

setColorOf:to:forCellAt::

± **setColorOf:**(int)*element* **to:**(NXColor)*aColor* **forCellAt**:(int)*xPos* :(int)*yPos*

Sets the color of the specified *element* for the cell at the matrix coordinates *xPos,yPos*. The argument *element* can be one of the following constants:

MISC_CV_CELLBACKGROUND
MISC_CV_CELLTEXT
MISC_CV_CELLHIGHLIGHT
MISC_CV_CELLTEXTHIGHLIGHT

See the section **Constants and Defined Types** for a description of the constants. Returns **self**.

See also: ± colorOf:forCellAt::, ± colorOf:, ± preserveCellColors:, ± setColorOf:to:

setColorTo:forDay:

± **setColorTo:**(NXColor)*aColor* **forDay:**(MiscCVDays)*aDay*

Sets the color of the header for the specified day. Returns **self**.

See also: ± colorForDay:

setDateDelegate:

± **setDateDelegate:***aDateObject*

Sets the date object used for day, date, and year calculations. It must conform to the **DateDelegate** protocol. Returns **self**.

See also: ± dateDelegate, ± updateDate:, ± displayDate:

setHighlightMode:

± **setHighlightMode:**(int)*mode*

Sets the highlight mode of the matrix. See documentation on **Matrix** for a description of what modes are available. Returns **self.**

See also: ± highlightMode:, setMode:(Matrix)

updateDate:

± **updateDate:***sender*

Updates the date stored in the date delegate by reading the day from matrix, and then redisplays the view. Returns **self.**

See also:± setDateDelegate:, ±dateDelegate, ± displayDate:

Protocols

The following protocol is necessary for any date delegate object:

```
@protocol DateDelegate

- (int)day;
- (int)month;
- (int)year;

- setYear:(int)year month:(int)month day:(int)day;

- (int)numberOfDaysInMonth;
- (int)startDayOfMonth;
- incrementMonth;
- decrementMonth;

- incrementYear;
- decrementYear;

- (const char *)monthStringValue;
- (const char *)dateStringValue;

@end
```

Constants and Defined Types

```
/* The days of the week */

enum MiscCVDaysEnum
{
    SUNDAY = 0,
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY
}

MISC_CV_CELLBACKGROUND          /* cell background color */
MISC_CV_CELLTEXT                /* cell text color */
MISC_CV_CELLHIGHLIGHT           /* cell background color when highlighted */
MISC_CV_CELLTEXTHIGHLIGHT       /* cell text color when highlighted */
MISC_CV_DOWHEADER               /* days of the week headers (S, M, T, ...) */
MISC_CV_MONTHHEADER             /* month header */
MISC_CV_YEARHEADER              /* year header */
MISC_CV_MONTHANDYEARHEADER      /* inline month and year header */
MISC_CV_INLINEMONTHHEADER       /* inline month header */
MISC_CV_INLINEYEARHEADER        /* inline year header */
```