Paul S. McCarthy and Eric Sunshine -- January 12, 1996

MiscTableCell

Inherits From: Cell

Declared In: MiscTableCell.h

Class Description

This is the cell class used by ;MiscTableScroll.rtf;;¬MiscTableScroll. It draws a one-pixel border along its bottom and right edges which produce one-pixel lines when displayed in a MiscTableScroll. This class is capable of displaying either text or an icon. This class provides a tag and separate text and background colors for normal and highlighted states. In addition to the -setIcon: and -icon methods which are inherited from Cell, this class provides support for unnamed images with the ;MiscTableCell.rtf;setImage:;¬-setImage: and ;MiscTableCell.rtf;image;¬-image methods. Finally, this class also supports the concept of an *owner*, and inherited font and color values.

Owner and Inheritence of Colors and Font

This class implements the concept of an *owner*. An owner provides the default font and colors used by the cells that it owns. This makes it fast and easy to change the appearance of an entire table: set the font and colors of the owner, and all of the cells that it owns automatically inherit the new font and colors. However, you can also set font and color values for individual cells. Values that you explicitly set supercede values implicitly inherited

from the owner. You can use this feature to emphasize special cells while still using inherited values for normal cells.

Another benefit of inherited values is reduced memory usage. Inherited values don't have to be stored in the individual cells. They just ask their owner for the current value whenever it is needed. This can lead to significant savings in large tables.

tc1_discussion;¬**Extensible Memory Management**

To minimize memory usage, storage for optional items is allocated on an as-needed basis. Memory is allocated only for the values that are actually set. If no optional items are set, then no extra memory at all is allocated.

The optional allocation scheme is extensible. Subclasses of MiscTableCell can extend this allocation scheme by appending their own dynamic variables to the end of the list used by this class. There is a strict ordering for the information stored in the variable length ;MiscTableCell.rtf;tc1_data;¬data. This ordering is enumerated by

the *;MiscTableCell.rtf;tc1_flags;¬tc1_flags*. Subclasses should append their data after the last of the information stored by this class. Methods are provided to determine whether or not a piece of information is stored in the data field as well as for finding its address. Convenience methods for extracting a given piece of information complete the interface.

The methods which return the address of the piece of data in *tc1_data* are dynamic in nature. For instance, text color is stored prior to background color in *tc1_data* so if both text color and background color have been set then the address of the storage for the background color will follow the text color. However, if text color is never set ± that is it is inherited from the owner ± and the background color has been set then the address of the background color will be at a different location. If text color gets set later then the storage for background color is shifted up to make room for the text color. Ergo, the strict ordering is maintained.

For example, code to locate the address of the highlighted background color is stored in the cell would look like this:

```
- (unsigned int) tc1HighlightBackgroundColorPos
```

```
{
unsigned int pos = [self tc1HighlightTextColorPos];
if (tc1_flags & MISC_TC1_SELF_TEXT_COLOR_H)
   pos += [self tc1HighlightTextColorLen];
return pos;
}
```

Since the highlighted background color is stored following the highlighted text color it first determines the location of the highlighted text color then adds the length of the highlighted text color to that address if the highlighted text color is in fact being stored locally. The actual shifting of the dynamic memory is done in the *-setUseOwner*... methods. For instance:

```
- setUseOwnerHighlightBackgroundColor: (BOOL) flag
   {
   if ([self useOwnerHighlightBackgroundColor] != flag)
      {
      unsigned int const pos = [self tc1HighlightBackgroundColorPos];
```

```
    unsigned int const len = [self tc1HighlightBackgroundColorLen];
    if (flag)
       {
       [self tc1DeleteDataPos:pos len:len];
       tc1_flags &= ~(MISC_TC1_SELF_BACKGROUND_COLOR_H);
       }
    else // (!flag)
       {
       NXColor color = [[self class] defaultHighlightBackgroundColor];
       [self tc1InsertData:&color pos:pos len:len];
       tc1_flags |= MISC_TC1_SELF_BACKGROUND_COLOR_H;
       }
    }
 return self;
 }
```

Finally, the *-setHighlightBackgroundColor:* method makes room for the data and then stores it:

```
- setHighlightBackgroundColor: (NXColor) c
   {
   NXColor* p;
   [self setUseOwnerHighlightBackgroundColor:NO];
   p = [self tc1HighlightBackgroundColorPtr];
   *p = c;
   return self;
   }
```

A subclass adding more data to the end of *tc1_data* would provide similar methods. Its counterpart to *-tc1HighlightBackgroundColorPos* would first call *-tc1HighlightBackgroundColorPos* to determine its address and then add the correct offset returned by *-tc1HighlightBackgroundColorLen* as appropriate.

Instance Variables

id **owner**;
int **tag**;
unsigned **tc1_flags**;
void* **tc1_data**;

owner	The owner of this cell; queried for font and color information.
tag	A general purpose slot for your use.
tc1_flags	Bit ;MiscTableCell.rtf;tc1_flags;¬flags describing the contents of **tc1_data**.
tc1_data;¬tc1_data	Variable length ;MiscTableCell.rtf;tc1_discussion;¬data which is allocated only as needed for storage of color values.

Method Types

Initializing, freeing, and copying	;MισχTαβλεXελλ.ρτφ;ινιτIχονXελλ:;← - initIconCell: ;MισχTαβλεXελλ.ρτφ;ινιτTεξτXελλ:;← - initTextCell: ;MισχTαβλεXελλ.ρτφ;φρεε;← - free ;MισχTαβλεXελλ.ρτφ;χοπψ;← - copyFromZone:
Drawing	;MισχTαβλεXελλ.ρτφ;χαλχXελλΣιζε:ινPεχτ:;← - calcCellSize:inRect: ;MισχTαβλεXελλ.ρτφ;δραωIνσιδε:ινςιεω:;← - drawInside:inView: ;MισχTαβλεXελλ.ρτφ;δραωΣελφ:ινςιεω:;← - drawSelf:inView: ;MισχTαβλεXελλ.ρτφ;βγXολορ;← - bgColor ;MισχTαβλεXελλ.ρτφ;φγXολορ;← - fgColor
Tag manipulation	;MισχTαβλεXελλ.ρτφ;σετTαγ:;← - setTag: ;MισχTαβλεXελλ.ρτφ;ταγ;← - tag

Font manipulation	;ΜισχΤαβλεΧελλ.ρτφ;φοντ;← - font ;ΜισχΤαβλεΧελλ.ρτφ;σετΦοντ:;← - setFont:
Icon	;ΜισχΤαβλεΧελλ.ρτφ;ιμαγε;← - image ;ΜισχΤαβλεΧελλ.ρτφ;σετΙμαγε:;← - setImage:
Setting and querying colors	;MiscTableCell.rtf;defaultBackgroundColor;¬ + defaultBackgroundColor ;MiscTableCell.rtf;defaultFont;¬ + defaultFont ;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬ + defaultHighlightBackgroundColor ;MiscTableCell.rtf;defaultHighlightTextColor;¬ + defaultHighlightTextColor ;MiscTableCell.rtf;defaultTextColor;¬ + defaultTextColor ;ΜισχΤαβλεΧελλ.ρτφ;βαχκγρουνδΧολορ;← - backgroundColor ;ΜισχΤαβλεΧελλ.ρτφ;βαχκγρουνδΓραψ;← - backgroundGray ;ΜισχΤαβλεΧελλ.ρτφ;ηιγηλιγητΒαχκγρουνδΧολορ;← -

highlightBackgroundColor
;ΜισχΤαβλεΧελλ.ρτφ;ηιγηλιγητΒαχκγρουνδΓραψ;← -
highlightBackgroundGray
;ΜισχΤαβλεΧελλ.ρτφ;ηιγηλιγητΤεξτΧολορ;← - highlightTextColor
;ΜισχΤαβλεΧελλ.ρτφ;ηιγηλιγητΤεξτΓραψ;← - highlightTextGray
;ΜισχΤαβλεΧελλ.ρτφ;σετΒαχκγρουνδΧολορ:;← - setBackgroundColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΒαχκγρουνδΓραψ:;← - setBackgroundGray:
;ΜισχΤαβλεΧελλ.ρτφ;σετΗιγηλιγητΒαχκγρουνδΧολορ:;← -
setHighlightBackgroundColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΗιγηλιγητΒαχκγρουνδΓραψ:;← -
setHighlightBackgroundGray:
;ΜισχΤαβλεΧελλ.ρτφ;σετΗιγηλιγητΤεξτΧολορ:;← - setHighlightTextColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΗιγηλιγητΤεξτΓραψ:;← - setHighlightTextGray:
;ΜισχΤαβλεΧελλ.ρτφ;σετΤεξτΧολορ:;← - setTextColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΤεξτΓραψ:;← - setTextGray:

	;ΜισχΤαβλεΧελλ.ρτφ;τεξτΧολορ;← - textColor ;ΜισχΤαβλεΧελλ.ρτφ;τεξτΓραψ;← - textGray
Setting and querying owner	;ΜισχΤαβλεΧελλ.ρτφ;οωνερ;← - owner ;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερ:;← - setOwner:
Colors and owner inheritence	;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερΒαχκγρουνδΧολορ:;← - setOwnerBackgroundColor: ;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερΦοντ:;← - setOwnerFont: ;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερΗιγηλιγητΒαχκγρουνδΧολορ:;← - setOwnerHighlightBackgroundColor: ;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερΗιγηλιγητΤεξτΧολορ:;← - setOwnerHighlightTextColor: ;ΜισχΤαβλεΧελλ.ρτφ;σετΟωνερΤεξτΧολορ:;← - setOwnerTextColor: ;ΜισχΤαβλεΧελλ.ρτφ;σετΥσεΟωνερΒαχκγρουνδΧολορ:;← -

setUseOwnerBackgroundColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΥσεΟωνερΦοντ:;← - setUseOwnerFont:
;ΜισχΤαβλεΧελλ.ρτφ;σετΥσεΟωνερΗιγηλιγητΒαχκγρουνδΧολορ:;← -
setUseOwnerHighlightBackgroundColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΥσεΟωνερΗιγηλιγητΤεξτΧολορ:;← -
setUseOwnerHighlightTextColor:
;ΜισχΤαβλεΧελλ.ρτφ;σετΥσεΟωνερΤεξτΧολορ:;← - setUseOwnerTextColor:
;ΜισχΤαβλεΧελλ.ρτφ;υσεΟωνερΒαχκγρουνδΧολορ;← -
useOwnerBackgroundColor
;ΜισχΤαβλεΧελλ.ρτφ;υσεΟωνερΦοντ;← - useOwnerFont
;ΜισχΤαβλεΧελλ.ρτφ;υσεΟωνερΗιγηλιγητΒαχκγρουνδΧολορ;← -
useOwnerHighlightBackgroundColor
;ΜισχΤαβλεΧελλ.ρτφ;υσεΟωνερΗιγηλιγητΤεξτΧολορ;← -
useOwnerHighlightTextColor
;ΜισχΤαβλεΧελλ.ρτφ;υσεΟωνερΤεξτΧολορ;← - useOwnerTextColor

Archiving	;ΜισχΤαβλεΧελλ.ρτφ;ρεαδ:;← - read: ;ΜισχΤαβλεΧελλ.ρτφ;ωριτε:;← - write:
Extensible conditional allocations	;ΜισχΤαβλεΧελλ.ρτφ;τχ1Βαχκγρουνδ ΧολορΛεν;← - tc1BackgroundColorLen ;ΜισχΤαβλεΧελλ.ρτφ;τχ1Βαχκγρουνδ ΧολορΠοσ;← - tc1BackgroundColorPos ;ΜισχΤαβλεΧελλ.ρτφ;τχ1Βαχκγρουνδ ΧολορΠτρ;← - tc1BackgroundColorPtr ;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΔαταΣιζε;← - tc1DataSize ;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΔελετεΔαταΠοσ:λεν:;← - tc1DeleteDataPos:len: ;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΔεστροψΔατα;← - tc1DestroyData ;ΜισχΤαβλεΧελλ.ρτφ;τχ1Φλαγσ;← - tc1Flags ;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΒαχκγρουνδ ΧολορΛεν;← - tc1HighlightBackgroundColorLen ;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΒαχκγρουνδ ΧολορΠοσ;← -

tc1HighlightBackgroundColorPos
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΒαχκγρουνδΧολορΠτρ;← -
tc1HighlightBackgroundColorPtr
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΤεξτΧολορΛεν;← -
tc1HighlightTextColorLen
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΤεξτΧολορΠοσ;← -
tc1HighlightTextColorPos
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΗιγηλιγητΤεξτΧολορΠτρ;← - tc1HighlightTextColorPtr
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΙνσερτΔατα:ποσ:λεν:;← - tc1InsertData:pos:len:
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΤεξτΧολορΛεν;← - tc1TextColorLen
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΤεξτΧολορΠοσ;← - tc1TextColorPos
;ΜισχΤαβλεΧελλ.ρτφ;τχ1ΤεξτΧολορΠτρ;← - tc1TextColorPtr

Class Methods

defaultBackgroundColor;¬defaultBackgroundColor
+ (NXColor)**defaultBackgroundColor**

Returns NX_COLORLTGRAY as the default background color. This is the color which is returned by **;MiscTableCell.rtf;backgroundColor;¬-backgroundColor** if a custom color has not been set for this cell and no owner has been set or the owner does not respond to the **-backgroundColor** message. This is also the color used upon receipt of a **;MiscTableCell.rtf;setUseOwnerBackgroundColor:;¬-setUseOwnerBackgroundColor:***NO* message. Subclasses should override this method if this color is inappropriate.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor, **;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬ + defaultHighlightBackgroundColor**, **;MiscTableCell.rtf;defaultHighlightTextColor;¬€+€defaultHighlightTextColor**, **;MiscTableCell.rtf;defaultTextColor;¬ + defaultTextColor**, **;MiscTableCell.rtf;setBackgroundColor:;¬ ±**

setBackgroundColor:, **;MiscTableCell.rtf;setUseOwnerBackgroundColor:;¬€±**
€setUseOwnerBackgroundColor:

defaultFont;¬defaultFont
+ **defaultFont**

Returns the user-font at point-size 12 as the default font. This is the font which is used when initializing the cell as a text cell with **;MiscTableCell.rtf;initTextCell:;¬±initTextCell:**. Subclasses should override this method if this value is inappropriate.

See also: ;MiscTableCell.rtf;font;¬ ± font, **;MiscTableCell.rtf;initTextCell:;¬ ± initTextCell:**, **;MiscTableCell.rtf;setFont:;¬ ± setFont:**

defaultHighlightBackgroundColor;¬defaultHighlightBackgroundColor
+ (NXColor)**defaultHighlightBackgroundColor**

Returns NX_COLORWHITE as the default highlighted background color. This is the color which is returned by **;MiscTableCell.rtf;highlightBackgroundColor;¬-highlightBackgroundColor** if a custom color has not been set for this cell and no owner has been set or the owner does not respond to the **-highlightBackgroundColor** message. This is also the color used upon receipt of a **;MiscTableCell.rtf;setUseOwnerHighlightBackgroundColor:;¬-setUseOwnerHighlightBackgroundColor:***NO* message. Subclasses should override this method if this color is inappropriate.

See also: ;MiscTableCell.rtf;defaultBackgroundColor;¬ + defaultBackgroundColor, **;MiscTableCell.rtf;defaultHighlightTextColor;¬€+€defaultHighlightTextColor**, **;MiscTableCell.rtf;defaultTextColor;¬ + defaultTextColor**, **;MiscTableCell.rtf;highlightBackgroundColor;¬€±€highlightBackgroundColor**, **;MiscTableCell.rtf;setHighlightBackgroundColor:;¬ ±**

setHighlightBackgroundColor:, **;MiscTableCell.rtf;setUseOwnerHighlightBackgroundColor:;¬€± €setUseOwnerHighlightBackgroundColor:**

defaultHighlightTextColor;¬defaultHighlightTextColor
+ (NXColor)**defaultHighlightTextColor**

Returns NX_COLORBLACK as the default highlighted text color. This is the color which is returned by **;MiscTableCell.rtf;highlightTextColor;¬-highlightTextColor** if a custom color has not been set for this cell and no owner has been set or the owner does not respond to the **-highlightTextColor** message. This is also the color used upon receipt of a **;MiscTableCell.rtf;setUseOwnerHighlightTextColor:;¬-setUseOwnerHighlightTextColor:***NO* message. Subclasses should override this method if this color is inappropriate.

See also: ;MiscTableCell.rtf;defaultBackgroundColor;¬ + **defaultBackgroundColor**,

;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬€+€defaultHighlightBackgroundColor, **;MiscTableCell.rtf;defaultTextColor;¬ + defaultTextColor**, ;MiscTableCell.rtf;highlightTextColor;¬€± €**highlightTextColor**, **;MiscTableCell.rtf;setHighlightBackgroundColor:;¬ ± setHighlightBackgroundColor:**, **;MiscTableCell.rtf;setUseOwnerHighlightBackgroundColor:;¬€± €setUseOwnerHighlightBackgroundColor:**

defaultTextColor;¬defaultTextColor

+ (NXColor)**defaultTextColor**

Returns NX_COLORBLACK as the default text color. This is the color which is returned by **;MiscTableCell.rtf;textColor;¬-textColor** if a custom color has not been set for this cell and no owner has been set or the owner does not respond to the **-textColor** message. This is also the color used upon receipt of a **;MiscTableCell.rtf;setUseOwnerTextColor:;¬-setUseOwnerTextColor:***NO* message. Subclasses should override this method if this color is inappropriate.

See also: **;MiscTableCell.rtf;defaultBackgroundColor;¬ + defaultBackgroundColor**, **;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬€+€defaultHighlightBackgroundColor**, **;MiscTableCell.rtf;defaultHighlightTextColor;¬€+€defaultHighlightTextColor**, **;MiscTableCell.rtf;setHighlightBackgroundColor:;¬ ± setHighlightBackgroundColor:**, **;MiscTableCell.rtf;setUseOwnerHighlightBackgroundColor:;¬€± €setUseOwnerHighlightBackgroundColor:**, **;MiscTableCell.rtf;textColor;¬ ± textColor**

Instance Methods

backgroundColor;¬backgroundColor

- (NXColor)**backgroundColor**

Returns the color that is used to draw the background of the cell in its normal (unhighlighted) state. This is the

color which has been set with **;MiscTableCell.rtf;setBackgroundColor:;¬±setBackgroundColor:** if it was ever called. If not, and the owner has been set and responds to **±backgroundColor** then it is queried and that color is returned. If both of the above fail then the value from **;MiscTableCell.rtf;defaultBackgroundColor;¬+defaultBackgroundColor** is returned.

See also: ;MiscTableCell.rtf;backgroundGray;¬ ± backgroundGray, **;MiscTableCell.rtf;defaultBackgroundColor;¬** + **defaultBackgroundColor**, **;MiscTableCell.rtf;highlightBackgroundColor;¬ ± highlightBackgroundColor**, **;MiscTableCell.rtf;highlightTextColor;¬€±€highlightTextColor**, **;MiscTableCell.rtf;setBackgroundColor:;¬ ± setBackgroundColor:**, **;MiscTableCell.rtf;textColor;¬ ± textColor**

backgroundGray;¬backgroundGray

- (float)**backgroundGray**

Returns the gray value from by **NXConvertColorToGray()** given the color returned by **;MiscTableCell.rtf;backgroundColor;¬-backgroundColor**.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor, **;MiscTableCell.rtf;highlightBackgroundGray;¬ ± highlightBackgroundGray**, **;MiscTableCell.rtf;highlightTextGray;¬ ± highlightTextGray**, **;MiscTableCell.rtf;setBackgroundGray:;¬€± €setBackgroundGray:**, **;MiscTableCell.rtf;textGray;¬ ± textGray**

bgColor;¬bgColor

- (NXColor)**bgColor**

Returns the color that is used to fill the background of the cell during drawing. The color returned by this method is dependant upon the cell's **state**. If **state** is 0 then this method calls **;MiscTableCell.rtf;backgroundColor;¬-backgroundColor** to derive the color, else it calls

;MiscTableCell.rtf;highlightBackgroundColor;¬-highlightBackgroundColor. Subclasses should override this method if they have different criteria for determining the cell's background color during drawing or if it is inappropriate to return the highlighted background color when state is non-zero.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor, **;MiscTableCell.rtf;fgColor;¬ ± fgColor**, **;MiscTableCell.rtf;highlightBackgroundColor;¬ ± highlightBackgroundColor**

calcCellSize:inRect:;¬calcCellSize:inRect:

- **calcCellSize:**(NXSize*)*theSize*
  inRect:(NXRect const*)*aRect*

Returns **self**, and by reference in *theSize* the minimum width and height required for displaying the cell in *aRect*.

See also: ± calcCellSize:inRect: (Cell)

copy;¬copyFromZone:

- **copyFromZone:**

Allocates, initializes, and returns a copy of the receiving cell. The copy is allocated from *zone* and is assigned the same contents as the receiver.

See also: ± copyFromZone: (Cell), **;MiscTableCell.rtf;free;¬ ± free**, **;MiscTableCell.rtf;initIconCell:;¬ ± initIconCell:**, **;MiscTableCell.rtf;initTextCell:;¬ ± initTextCell:**

drawInside:inView:;¬drawInside:inView:

- **drawInside:**(NXRect const*)*cellFrame*
  inView: *controlView*

Draws the inside of the cell, but not the border, in *cellFrame* within *controlView*. *cellFrame* should be the same rectangle passed to ;MiscTableCell.rtf;drawSelf:inView:;¬-**drawSelf:inView:**. The PostScript focus must be locked on *controlView* when this message is sent. Returnse **self**.

See also: - **drawInside:inView:** (Cell), ;ΜισχΤαβλεΧελλ.ρτφ;δραωΣελφ:ινςιεω:;← - **drawSelf:inView:**, - **lockFocus** (View)

drawSelf:inView:;¬drawSelf:inView:

- **drawSelf:**(NXRect const*)*cellFrame*
  inView: *controlView*

Displays the cell in *cellFrame* within *controlView*. The PostScript focus must be locked on *controlView* when this message is sent. Draws the border of the cell, then invokes ;MiscTableCell.rtf;drawInside:inView:;¬±**drawInside:inView:**. Returns **self**.

See also: ;ΜισχΤαβλεΧελλ.ρτφ;δραωΙνσιδε:ινζιεω:;← **- drawInside:inView:**, ± **drawSelf:inView:** (Cell), **- lockFocus** (View)

fgColor;¬fgColor

- (NXColor)**fgColor**

Returns the color that is used for the text in the cell during drawing. The color returned by this method is dependant upon the cell's **state**. If **state** is 0 then this method calls **;MiscTableCell.rtf;textColor;¬-textColor** to derive the color, else it calls **;MiscTableCell.rtf;highlightTextColor;¬-highlightTextColor**. Subclasses should override this method if they have different criteria for determining the cell's text color during drawing or if it is inappropriate to return the highlighted text color when state is non-zero.

See also: ;MiscTableCell.rtf;bgColor;¬ ± bgColor, **;MiscTableCell.rtf;highlightTextColor;¬ ± highlightTextColor**, **;MiscTableCell.rtf;textColor;¬ ± textColor**

font;¬font

- **font**

Returns the Font used to display text in the cell. Returns **nil** if the receiver isn't a text cell.

See also: ;MiscTableCell.rtf;defaultFont;¬ + defaultFont, **;MiscTableCell.rtf;setFont:;¬ ± setFont:**

free;¬free

- **free**

Frees the memory used by the cell and returns **nil**.

See also: ;MiscTableCell.rtf;copyFromZone:;¬ ± copyFromZone:, **;MiscTableCell.rtf;initIconCell:;¬ ±**

initIconCell:, **;MiscTableCell.rtf;initTextCell:;¬ ± initTextCell:**

highlightBackgroundColor;¬highlightBackgroundColor

- (NXColor)**highlightBackgroundColor**

Returns the color that is used to draw the background of the cell in its highlighted state. This is the color which has been set with **;MiscTableCell.rtf;setHighlightBackgroundColor:;¬±setHighlightBackgroundColor:** if it was ever called. If not, and the owner has been set and responds to **±highlightBackgroundColor** then it is queried and that color is returned. If both of the above fail then the value from **;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬+defaultHighlightBackgroundColor** is returned.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor, **;MiscTableCell.rtf;defaultHighlightBackgroundColor;¬€+€defaultHighlightBackgroundColor**, **;MiscTableCell.rtf;highlightTextColor;¬€±€highlightTextColor**,

;MiscTableCell.rtf;highlightBackgroundGray;¬€±€highlightBackgroundGray,
;MiscTableCell.rtf;setHighlightBackgroundColor:;¬ ± setHighlightBackgroundColor:,
;MiscTableCell.rtf;textColor;¬€± textColor

highlightBackgroundGray;¬highlightBackgroundGray

- (float)**highlightBackgroundGray**

Returns the gray value from by **NXConvertColorToGray()** given the color returned by **;MiscTableCell.rtf;highlightBackgroundColor;¬-highlightBackgroundColor**.

See also: **;MiscTableCell.rtf;backgroundGray;¬ ± backgroundGray**,
;MiscTableCell.rtf;highlightBackgroundColor;¬ ± highlightBackgroundColor,
;MiscTableCell.rtf;highlightTextGray;¬ ± highlightTextGray,
;MiscTableCell.rtf;setHighlightBackgroundGray:;¬€±€setHighlightBackgroundGray:,

;MiscTableCell.rtf;textGray;¬ ± textGray

highlightTextColor;¬highlightTextColor

- (NXColor)**highlightTextColor**

Returns the color that is used to draw the text in the cell in its highlighted state. This is the color which has been set with **;MiscTableCell.rtf;setHighlightTextColor:;¬±setHighlightTextColor:** if it was ever called. If not, and the owner has been set and responds to **±highlightTextColor** then it is queried and that color is returned. If both of the above fail then the value from **;MiscTableCell.rtf;defaultHighlightTextColor;¬+defaultHighlightTextColor** is returned.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor, **;MiscTableCell.rtf;defaultHighlightTextColor;¬€+€defaultHighlightTextColor**, **;MiscTableCell.rtf;highlightBackgroundColor;¬€±€highlightBackgroundColor**,

;MiscTableCell.rtf;highlightTextGray;¬€±€highlightTextGray, **;MiscTableCell.rtf;setHighlightTextColor:;¬ ± setHighlightTextColor:**, **;MiscTableCell.rtf;textColor;¬€± textColor**

highlightTextGray;¬highlightTextGray

- (float)**highlightTextGray**

Returns the gray value from by **NXConvertColorToGray()** given the color returned by **;MiscTableCell.rtf;highlightTextColor;¬-highlightTextColor**.

See also: ;MiscTableCell.rtf;backgroundGray;¬ ± backgroundGray, **;MiscTableCell.rtf;highlightBackgroundGray;¬ ± highlightBackgroundGray**, **;MiscTableCell.rtf;highlightTextColor;¬ ± highlightTextColor**, **;MiscTableCell.rtf;setHighlightTextGray:;¬€± €setHighlightTextGray:**, **;MiscTableCell.rtf;textGray;¬ ± textGray**

image;¬image

- **image**

Returns a pointer to an NXImage object if the cell is an icon cell, otherwise returns 0.

See also: **± icon** (Cell), **;MiscTableCell.rtf;initIconCell:;¬ ± initIconCell:**, **± setIcon:** (Cell), **;MiscTableCell.rtf;setImage:;¬ ± setImage:**

initIconCell:;¬initIconCell:

- **initIconCell:**(char const*)*s*

Prepare a newly allocated cell that will display an icon. *S* is the name of the image.

See also:

initTextCell:;¬initTextCell:

- **initTextCell:**(char const*)*s*

Prepare a newly allocated cell that will display text. The font and colors will use inherited or default values until explicitly set.

See also:

owner;¬owner

- **owner**

Returns the **owner** of the cell, or 0 if there is none.

See also:

read:;¬read:

- **read:**(NXTypedStream*)*stream*

Unarchives a cell from *stream*, that was archived with -**write:**.

See also: **-write:**

setBackgroundColor:;¬setBackgroundColor:

- **setBackgroundColor:**(NXColor)*c*

Sets the color that the cell will use to fill its background. This color will override any inherited value.

See also:

setBackgroundGray:;¬setBackgroundGray:

- **setBackgroundGray:**(float)*value*

The gray value is converted to a color value which is passed to **-setBackgroundColor:**.

See also:

setFont:;¬setFont:

- **setFont:***font*

Sets the font of a text cell to *font*. This will override any inherited value.

See also:

setHighlightBackgroundColor:;¬setHighlightBackgroundColor:

- **setHighlightBackgroundColor:**(NXColor)*c*

Sets the color that will be used to fill the background of a text cell when the cell is highlighted. This will override any inherited value.

See also:

setHighlightBackgroundGray:;¬setHighlightBackgroundGray:

- **setHighlightBackgroundGray:**(float)*value*

The gray value is converted to a color value which is passed to **-setHighlightBackgroundColor:**.

See also:

setHighlightTextColor:;¬setHighlightTextColor:

- **setHighlightTextColor:**(NXColor)*c*

Sets the color that will be used to draw the text of a text cell. This will override any inherited value.

See also:

setHighlightTextGray:;¬setHighlightTextGray:

- **setHighlightTextGray:**(float)*value*

The gray value is converted to a color value which is passed to **-setHighlightTextColor:**.

See also:

setImage:;¬setImage:

- **setImage:** *image*

Sets the image that will be displayed in an icon cell.

See also: **± icon** (Cell), **;MiscTableCell.rtf;image;¬ ± image**, **;MiscTableCell.rtf;initIconCell:;¬ ± initIconCell:**, **± setIcon:** (Cell)

setOwner:;¬setOwner:

- **setOwner:***obj*

Sets the **owner** of the cell to *obj*. The cell will inherit font and color values from *obj*.

See also:

setOwnerBackgroundColor:;¬setOwnerBackgroundColor:

- **setOwnerBackgroundColor:**(NXColor)*c*

Informs the cell that the owner's **backgroundColor** value has changed to ***c.*** The current implementation does nothing, since inherited owner values are not stored in the cells. However, the MiscTableScroll class tests for, and sends this message in preference to the **-setBackgroundColor:** message when the table distributes colors to the cells, so that cells can distinguish between cell-specific color assignments and global color assignments.

See also:

setOwnerFont:;¬setOwnerFont:

- **setOwnerFont:***font*

Informs the cell that the owner's **font** value has changed to *font*. If the cell is using the font value inherited from its **owner**, it will change its font in response to this message. Otherwise, it will ignore the message. The MiscTableScroll class tests for, and sends this message in preference to the **-setFont:** message when the table distributes fonts to the cells, so that cells can distinguish between cell-specific font assignments and global font assignments.

See also:

setOwnerHighlightBackgroundColor:;¬setOwnerHighlightBackgroundColor:

- **setOwnerHighlightBackgroundColor:**(NXColor)*c*

Informs the cell that the owner's **highlightBackgroundColor** value has changed to *c.* The current implementation does nothing, since inherited owner values are not stored in the cells. However, the MiscTableScroll class tests for, and sends this message in preference to the **-setHighlightBackgroundColor:** message when the table distributes colors to the cells, so that cells can distinguish between cell-specific color assignments and global color assignments.

See also:

setOwnerHighlightTextColor:;¬setOwnerHighlightTextColor:

- **setOwnerHighlightTextColor:**(NXColor)*c*

Informs the cell that the owner's **highlightTextColor** value has changed to *c.* The current implementation does nothing, since inherited owner values are not stored in the cells. However, the MiscTableScroll class tests for,

and sends this message in preference to the **-setHighlightTextColor:** message when the table distributes colors to the cells, so that cells can distinguish between cell-specific color assignments and global color assignments.

See also:

setOwnerTextColor:;¬setOwnerTextColor:

- **setOwnerTextColor:**(NXColor)*c*

Informs the cell that the owner's **textColor** value has changed to *c.* The current implementation does nothing, since inherited owner values are not stored in the cells. However, the MiscTableScroll class tests for, and sends this message in preference to the **-setTextColor:** message when the table distributes colors to the cells, so that cells can distinguish between cell-specific color assignments and global color assignments.

See also:

setTag:;¬setTag:

- **setTag:**(int)*x*

Sets the cell's **tag** value to x.

See also: ± tag

setTextColor:;¬setTextColor:

- **setTextColor:**(NXColor)*c*

Causes the cell to render its text with the color, *c*. This overrides any **textColor** value inherited from the **owner**.

See also:

setTextGray:;¬setTextGray:

- **setTextGray:**(float)*value*

Value is converted to a color value which is passed to **-setTextColor:**

See also:

setUseOwnerBackgroundColor:;¬setUseOwnerBackgroundColor:

- **setUseOwnerBackgroundColor:**(BOOL)*flag*

If *flag* is YES, the cell will discard any **backgroundColor** value that it is currently storing, and begin using the backgroundColor value provided by the **owner**. If the owner has not been set, or the owner does not respond to the -**backgroundColor** message, then the value returned by **+defaultBackgroundColor** will be used. If *flag* is

NO, the cell will make room to store a **backgroundColor** value, and initialize it to the value returned by the **+defaultBackgroundColor** method.

See also:

setUseOwnerFont:;¬setUseOwnerFont:

- **setUseOwnerFont:**(BOOL)*flag*

If *flag* is YES, the cell will use the **font** value inherited from its **owner**. If *flag* is NO, the cell will continue using its current font, or fonts set via the **-setFont** message rather than the font inherited from the owner. This method also updates the `MISC_TC1_SELF_FONT` bit of the **tc1_flags** bit mask.

See also:

setUseOwnerHighlightBackgroundColor:;¬setUseOwnerHighlightBackgroundColor:

- **setUseOwnerHighlightBackgroundColor:**(BOOL)*flag*

If *flag* is YES, the cell will discard any **highlightBackgroundColor** value that it is currently storing, and begin using the highlightBackgroundColor value provided by the **owner**. If the owner has not been set, or the owner does not respond to the -**highlightBackgroundColor** message, then the value returned by **+defaultHighlightBackgroundColor** will be used. If *flag* is NO, the cell will make room to store a **highlightBackgroundColor** value, and initialize it to the value returned by the **+defaultHighlightBackgroundColor** method.

See also:

setUseOwnerHighlightTextColor:;¬setUseOwnerHighlightTextColor:

- **setUseOwnerHighlightTextColor:**(BOOL)*flag*

If *flag* is YES, the cell will discard any **highlightTextColor** value that it is currently storing, and begin using the highlightColor value provided by the **owner**. If the owner has not been set, or the owner does not respond to the -**highlightTextColor** message, then the value returned by **+defaultHighlightTextColor** will be used. If *flag* is NO, the cell will make room to store a **highlightTextColor** value, and initialize it to the value returned by the **+defaultHighlightTextColor** method.

See also:

setUseOwnerTextColor:;¬setUseOwnerTextColor:

- **setUseOwnerTextColor:**(BOOL)*flag*

If *flag* is YES, the cell will discard any **textColor** value that it is currently storing, and begin using the textColor value provided by the **owner**. If the owner has not been set, or the owner does not respond to the -**textColor** message, then the value returned by **+defaultTextColor** will be used. If *flag* is NO, the cell will make room to

store a **textColor** value, and initialize it to the value returned by the **+defaultTextColor** method.

See also:

tag;¬tag

- (int)**tag**

Returns the cell's **tag** value.

See also: ;MiscTableCell.rtf;setTag:;¬ ± setTag:

tc1BackgroundColorLen;¬tc1BackgroundColorLen

- (unsigned int)**tc1BackgroundColorLen**

Returns the length of the **backgroundColor** value in the variable-length **tc1_data** field. This is currently sizeof(NXColor).

See also:

tc1BackgroundColorPos;¬tc1BackgroundColorPos

- (unsigned int)**tc1BackgroundColorPos**

Returns the current offset of the **backgroundColor** value in the variable-length **tc1_data** field. This value changes whenever preceeding values are added or removed.

See also:

tc1BackgroundColorPtr;¬tc1BackgroundColorPtr

- (NXColor*)**tc1BackgroundColorPtr**

Returns a pointer to the **backgroundColor** value in the variable-length **tc1_data** field. The value returned is only valid if **-useOwnerBackgroundColor** returns NO.

See also:

tc1DataSize;¬tc1DataSize

- (unsigned int)**tc1DataSize**

Returns the current total size of the variable-length **tc1_data** field. This value changes as optional values are added and removed.

See also:

tc1DeleteDataPos:len:;¬tc1DeleteDataPos:len:

- (void)**tc1DeleteDataPos:**(unsigned int)*pos*
  len:(unsigned int)*len*

Deletes *len* bytes at offset *pos* from the variable-length **tc1_data** field. All following data is shifted down by *len* bytes. This method is called internally whenever an optional value no longer needs to be stored because the inherited value from the owner is going to be used.

See also: -tc1InsertData:pos:len:

tc1DestroyData;¬tc1DestroyData

- (void)**tc1DestroyData**

This method is provided as a hook for subclasses that need to perform special actions (destructors) before the variable-length **tc1_data** field is freed. The MiscTableCell implementation of this class clears the **tc1_flags** bit mask to zero.

See also: -tc1FreeData

tc1Flags;¬tc1Flags

- (unsigned int)**tc1Flags**

Returns the current value of the **tc1_flags** field. This is a bit-mask indicating which optional values are currently stored in the variable-length **tc1_data** field. The flags also indicate whether or not the corresponding values inherited from the owner are used, since any value that has been set (and stored) in the cell overrides the inherited value. The values of the individual bits are declared in <MiscTableCell.h> as follows:

```
tc1_flags;¬#define MISC_TC1_HAS_TAG      (1 << 0)  /* obsolete */
#define MISC_TC1_SELF_FONT               (1 << 1)
#define MISC_TC1_SELF_TEXT_COLOR         (1 << 2)
#define MISC_TC1_SELF_BACKGROUND_COLOR   (1 << 3)
#define MISC_TC1_SELF_TEXT_COLOR_H       (1 << 4)
#define MISC_TC1_SELF_BACKGROUND_COLOR_H (1 << 5)
#define MISC_TC1_LAST_BIT                (1 << 5)
```

See also:

tc1DestroyData;¬tc1FreeData

- (void)**tc1FreeData**

Frees the variable-length **tc1_data** variable. This method is invoked from within the **-free** method, and also

within the **-read:** method.

See also: **-tc1DestroyData**

tc1HighlightBackgroundColorLen;¬tc1HighlightBackgroundColorLen

- (unsigned int)**tc1HighlightBackgroundColorLen**

Returns the length of the **highlightBackgroundColor** value in the variable-length **tc1_data** field. This is currently sizeof(NXColor).

See also:

tc1HighlightBackgroundColorPos;¬tc1HighlightBackgroundColorPos

- (unsigned int)**tc1HighlightBackgroundColorPos**

Returns the current offset of the **highlightBackgroundColor** value in the variable-length **tc1_data** field. This value changes whenever preceeding values are added or removed.

See also:

tc1HighlightBackgroundColorPtr;¬tc1HighlightBackgroundColorPtr

- (NXColor*)**tc1HighlightBackgroundColorPtr**

Returns a pointer to the **highlightBackgroundColor** value in the variable-length **tc1_data** field. The value returned is only valid if **-useOwnerHighlightBackgroundColor** returns NO.

See also:

tc1HighlightTextColorLen;¬tc1HighlightTextColorLen

- (unsigned int)**tc1HighlightTextColorLen**

Returns the length of the **highlightTextColor** value in the variable-length **tc1_data** field. This is currently sizeof(NXColor).

See also:

tc1HighlightTextColorPos;¬tc1HighlightTextColorPos

- (unsigned int)**tc1HighlightTextColorPos**

Returns the current offset of the **highlightTextColor** value in the variable-length **tc1_data** field. This value changes whenever preceeding values are added or removed.

See also:

tc1HighlightTextColorPtr;¬tc1HighlightTextColorPtr

- (NXColor*)**tc1HighlightTextColorPtr**

Returns a pointer to the **highlightTextColor** value in the variable-length **tc1_data** field. The value returned is only valid if **-useOwnerHighlightTextColor** returns NO.

See also:

tc1InsertData:pos:len:;¬tc1InsertData:pos:len:

- (void*)**tc1InsertData:**(void const*)*data*
  pos:(unsigned int)*pos*

len:(unsigned int)*len*

Inserts *data* at offset *pos* in the variable-length **tc1_data** field, shifting all following data by *len* bytes. This method is used internally to allocate the storage for optional values that do not already have storage allocated.

See also: -tc1DeleteDataPos:len:

tc1TextColorLen;¬tc1TextColorLen

- (unsigned int)**tc1TextColorLen**

Returns the length of the **textColor** value in the variable-length **tc1_data** field. This is currently sizeof(NXColor).

See also:

tc1TextColorPos;¬tc1TextColorPos

- (unsigned int)**tc1TextColorPos**

Returns the current offset of the **textColor** value in the variable-length **tc1_data** field. This offset changes as other values are set or cleared.

See also:

tc1TextColorPtr;¬tc1TextColorPtr

- (NXColor*)**tc1TextColorPtr**

Returns a pointer to the location of the **textColor** value stored in the variable-length **tc1_data** field. The value returned is only valid if **-useOwnerTextColor** returns NO.

See also:

textColor;¬textColor

- (NXColor)**textColor**

Returns the color that is used to draw the text in the cell in its normal (unhighlighted) state. This is the color which has been set with **;MiscTableCell.rtf;setTextColor:;¬±setTextColor:** if it was ever called. If not, and the owner has been set and responds to **±textColor** then it is queried and that color is returned. If both of the above fail then the value from **;MiscTableCell.rtf;defaultTextColor;¬+defaultTextColor** is returned.

See also: ;MiscTableCell.rtf;backgroundColor;¬ ± backgroundColor,**;MiscTableCell.rtf;defaultTextColor;¬** + **defaultTextColor**, **;MiscTableCell.rtf;highlightBackgroundColor;¬ ± highlightBackgroundColor**, **;MiscTableCell.rtf;highlightTextColor;¬€±€highlightTextColor**, **;MiscTableCell.rtf;setTextColor:;¬ ±**

setTextColor:, **;MiscTableCell.rtf;textGray;¬ ± textGray**

textGray;¬textGray

- (float)**textGray**

Returns the gray value from by **NXConvertColorToGray()** given the color returned by **;MiscTableCell.rtf;textColor;¬-textColor**.

See also: ;MiscTableCell.rtf;backgroundGray;¬ ± backgroundGray, **;MiscTableCell.rtf;highlightBackgroundGray;¬ ± highlightBackgroundGray**, **;MiscTableCell.rtf;highlightTextGray;¬ ± highlightTextGray**, **;MiscTableCell.rtf;setTextGray:;¬€± €setTextGray:**, **;MiscTableCell.rtf;textColor;¬ ± textColor**

useOwnerBackgroundColor;¬useOwnerBackgroundColor

- (BOOL)**useOwnerBackgroundColor**

Returns YES if the cell uses the **backgroundColor** inherited from its **owner**, otherwise NO.

See also: -setBackgroundColor:, -setUseOwnerBackgroundColor:

useOwnerFont;¬useOwnerFont

- (BOOL)**useOwnerFont**

Returns YES if the cell uses the **font** inherited from its **owner**, otherwise NO.

See also: -setFont:, -setUseOwnerFont:

useOwnerHighlightBackgroundColor;¬useOwnerHighlightBackgroundColor

- (BOOL)**useOwnerHighlightBackgroundColor**

Returns YES if the cell uses the **highlightBackgroundColor** inherited from its **owner**, otherwise NO.

See also: -setHighlightBackgroundColor:, -setUseOwnerHighlightBackgroundColor:

useOwnerHighlightTextColor;¬useOwnerHighlightTextColor

- (BOOL)**useOwnerHighlightTextColor**

Returns YES if the cell uses the **highlightTextColor** inherited from its **owner**, otherwise NO.

See also: -setHighlightTextColor:, -setUseOwnerHighlightTextColor:

useOwnerTextColor;¬useOwnerTextColor

- (BOOL)**useOwnerTextColor**

Returns YES if the cell uses the **textColor** value inherited from its **owner**, otherwise NO.

See also: **-setTextColor:, -setUseOwnerTextColor:**

write:;¬write:

- **write:**(NXTypedStream*)*stream*

Archives the cell on *stream* so that it can be unarchived with **-read:**.

See also: **-read:**

Constants and Defined Types

```
tc1_flags;¬#define MISC_TC1_HAS_TAG      (1 << 0)  /* obsolete */
#define MISC_TC1_SELF_FONT               (1 << 1)
#define MISC_TC1_SELF_TEXT_COLOR         (1 << 2)
#define MISC_TC1_SELF_BACKGROUND_COLOR   (1 << 3)
#define MISC_TC1_SELF_TEXT_COLOR_H       (1 << 4)
#define MISC_TC1_SELF_BACKGROUND_COLOR_H (1 << 5)
#define MISC_TC1_LAST_BIT                (1 << 5)
```