

# **FC\_Display**

Olivier Laviale 2002

**COLLABORATORS**

	<i>TITLE :</i> FC_Display		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Olivier Laviale 2002	August 24, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>FC_Display</b>	<b>1</b>
1.1	Feelin : FC_Display . . . . .	1
1.2	FC_Display / FM_CreatePen . . . . .	2
1.3	FC_Display / FM_DeletePen . . . . .	2
1.4	FC_Display / FA_Display_ColorMap . . . . .	2
1.5	FC_Display / FA_Display_Penlist . . . . .	3
1.6	FC_Display / FA_Display_Scheme . . . . .	3
1.7	FC_Display / FA_Pen_Dark . . . . .	3
1.8	FC_Display / FA_Pen_Fill . . . . .	4
1.9	FC_Display / FA_Pen_HalfDark . . . . .	4
1.10	FC_Display / FA_Pen_HalfShadow . . . . .	5
1.11	FC_Display / FA_Pen_HalfShine . . . . .	5
1.12	FC_Display / FA_Pen_Highlight . . . . .	5
1.13	FC_Display / FA_Pen_Shadow . . . . .	6
1.14	FC_Display / FA_Pen_Shine . . . . .	6
1.15	FC_Display / FA_Pen_Text . . . . .	6
1.16	FC_Display / FA_Screen . . . . .	7
1.17	FC_Display / FA_Screen_Depth . . . . .	7
1.18	FC_Display / FeelinPen . . . . .	7

---

# Chapter 1

## FC\_Display

### 1.1 Feelin : FC\_Display

FC\_Display

This class is designed to manage display environnements that may be shared by different applications. Applications use a DisplayContext object as an interface to Display objects.

Each Display object manages its own screen (a special one handles Workbench screen), screen's colors and ressources.

You should never use FC\_Display to create displays, but rather use the DisplayServer to open / create them.

of an application (screen, colors...)

Its first goal is to create a default palette of colors used by every objects to render. By default this palette will be created from screen's pens (SHINEPEN, FILLPEN, SHADOWPEN) but you can create your own scheme using FA\_Pen\_Shine, FA\_Pen\_Fill and FA\_Pen\_Dark attributes on object creation time. You can also modify text color using FA\_Pen\_Text and highlight color using FA\_Pen\_Highlight. Each application can have its own color scheme if you want to.

An instance of this class is created by every Client objects. Use client's instance if you need to allocate some colors, because creating a new Display objet will be a waste of time and ressources.

As usual, every colors allocated using FM\_CreatePen are freed when Display object is freed.

Methods

**FM\_CreatePen FM\_DeletePen**

FM\_Display\_CreateScheme

FM\_Display\_DeleteScheme

Attributes

**FA\_Display\_ColorMap FA\_Display\_Penlist**

**FA\_Display\_Scheme**

**FA\_Pen\_Shine FA\_Pen\_HalfShine**

**FA\_Pen\_Fill FA\_Pen\_HalfShadow**

**FA\_Pen\_Shadow FA\_Pen\_HalfDark**

**FA\_Pen\_Dark FA\_Pen\_Text**

**FA\_Pen\_Highlight**

**FA\_Screen FA\_Screen\_Depth**

## 1.2 FC\_Display / FM\_CreatePen

NAME

FM\_CreatePen -- (00.00)

SYNOPSIS

F\_DoA(obj,FM\_CreatePen,[penspec])

FUNCTION

This method can be used by objects to figure out what pen to use to represent a given color.

The method will try to find the color in the Display closest to the specified color. If there is no color within a good tolerance, then a new one will be allocated, if available. If none is available, then the closest one found will be returned.

INPUTS

penspec - pointer to a FeelinPenSpec structure holding a string describing the color to create:

s:<n> - where <n> is a color from the color scheme. e.g. 's:1' will be [FA\\_Pen\\_Shine](#) . p:<n> - where <n> is a pen number from the Display's colormap. e.g. 'p:3' will be pen 3 of the colormap. r:<r,g,b> - where <r>, <g> and <b> are hexadecimal values of the color to create. e.g. 'r:FF0000' will be full red.

RESULT

A pen value. Don't forget to release the color using [FM\\_DeletePen](#) .

SEE ALSO

[FM\\_DeletePen](#)

## 1.3 FC\_Display / FM\_DeletePen

NAME

FM\_DeletePen -- (00.00)

SYNOPSIS

F\_DoA(obj,FM\_DeletePen,[pen])

FUNCTION

Return the palette entry for use by other objects. If the reference count for this palette entry goes to zero, then it may be reset to another RGB value.

INPUTS

pen, a pen value returned by [FM\\_CreatePen](#).

## 1.4 FC\_Display / FA\_Display\_ColorMap

NAME

FA\_Display\_ColorMap -- (00.00) [..G], PTR TO ColorMap

FUNCTION

Colormap of the current display context.

SEE ALSO

[FM\\_CreatePen](#)

[FA\\_Screen](#)

---

## 1.5 FC\_Display / FA\_Display\_Penlist

NAME

FA\_Display\_Penlist -- (00.00) [..G], PTR TO FeelinList

FUNCTION

This attribute returns a pointer to the FeelinList structure where allocated pens are linked to.

SEE ALSO

[FA\\_Display\\_Scheme](#)

## 1.6 FC\_Display / FA\_Display\_Scheme

NAME

FA\_Display\_Scheme -- (00.00) [..G], PTR TO INT

FUNCTION

This attribute returns a pointer to an array of long words representing the color scheme.

When Display object is created a scheme of colors is built from the current display context. This scheme is made of 9 colors : Text, Shine, HalfShine, Fill, HalfShadow, Shadow, HalfDark, Dark, Highlight. These pens are used by every objects to render.

By default, Shine, Fill, Dark, Text and Highlight are gotten from the current display context, others are computed from Shine, Fill and Dark. You can also specify pens to use to build the scheme using attributes FA\_Pen\_Shine, FA\_Pen\_Fill and FA\_Pen\_Dark. Text and Highlight pens can also be chosen using FA\_Pen\_Text and FA\_Pen\_Highlight attributes.

Note that these attributes should be defined by the user and not the programmer.

NOTE

Color scheme is a simple array of long words. You should use FV\_Pen\_Xxx values to get colors from this array e.g. dark := scheme[FV\_Pen\_Dark]. In the array, Pens are ordered as follow :

FV\_Pen\_Text

FV\_Pen\_Shine FV\_Pen\_HalfShine FV\_Pen\_Fill FV\_Pen\_HalfShadow FV\_Pen\_Shadow FV\_Pen\_HalfDark FV\_Pen\_Dark

FV\_Pen\_Highlight

SEE ALSO

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_HalfShine](#) [FA\\_Pen\\_Highlight](#)

[FA\\_Pen\\_Shadow](#) [FA\\_Pen\\_Shine](#)

[FA\\_Pen\\_Text](#)

## 1.7 FC\_Display / FA\_Pen\_Dark

NAME

FA\_Pen\_Dark -- (00.00) [I.G], LONG

FUNCTION

This attribute returns the color register used as Dark.

It can also be supplied when creating a Display object or a new scheme (using FM\_Display\_CreateScheme) to alter scheme building. See [FM\\_CreatePen](#) to know about pen specifications.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Fill](#) [FA\\_Pen\\_HalfDark](#)

[FA\\_Pen\\_HalfShadow](#) [FA\\_Pen\\_HalfShine](#)

[FA\\_Pen\\_Highlight](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

## 1.8 FC\_Display / FA\_Pen\_Fill

NAME

FA\_Pen\_Fill -- (00.00) [I.G], LONG

FUNCTION

This attribute returns the color register used as Fill.

It can also be supplied when creating a Display object or a new scheme (using FM\_Display\_CreateScheme) to alter scheme building. See [FM\\_CreatePen](#) to know about pen specifications.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_HalfDark](#)

[FA\\_Pen\\_HalfShadow](#) [FA\\_Pen\\_HalfShine](#)

[FA\\_Pen\\_Highlight](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

## 1.9 FC\_Display / FA\_Pen\_HalfDark

NAME

FA\_Pen\_HalfDark -- (00.00) [..G], LONG

FUNCTION

This attribute returns the color register used as HalfDark.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfShadow](#) [FA\\_Pen\\_HalfShine](#)

[FA\\_Pen\\_Highlight](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

---

## 1.10 FC\_Display / FA\_Pen\_HalfShadow

NAME

FA\_Pen\_HalfShadow -- (00.00) [..G], LONG

FUNCTION

This attribute returns the color register used as HalfShadow.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShine](#)

[FA\\_Pen\\_Highlight](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

## 1.11 FC\_Display / FA\_Pen\_HalfShine

NAME

FA\_Pen\_HalfShine -- (00.00) [..G], LONG

FUNCTION

This attribute returns the color register used as HalfShine.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_Highlight](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

## 1.12 FC\_Display / FA\_Pen\_Highlight

NAME

FA\_Pen\_Highlight -- (00.00) [I.G], LONG

FUNCTION

This attribute returns the color register used as Highlight.

It can also be supplied when creating a Display object or a new scheme (using [FM\\_Display\\_CreateScheme](#)) to alter scheme building. See [FM\\_CreatePen](#) to know about pen specifications.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_HalfShine](#) [FA\\_Pen\\_Shadow](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

---



### 1.13 FC\_Display / FA\_Pen\_Shadow

NAME

FA\_Pen\_Shadow -- (00.00) [..G],

FUNCTION

This attribute returns the color register used as Shadow.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_HalfShine](#) [FA\\_Pen\\_Highlight](#)

[FA\\_Pen\\_Shine](#) [FA\\_Pen\\_Text](#)

### 1.14 FC\_Display / FA\_Pen\_Shine

NAME

FA\_Pen\_Shine -- (00.00) [I.G], LONG

FUNCTION

This attribute returns the color register used as Shine.

It can also be supplied when creating a Display object or a new scheme (using [FM\\_Display\\_CreateScheme](#)) to alter scheme building. See [FM\\_CreatePen](#) to know about pen specifications.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_HalfShine](#) [FA\\_Pen\\_Highlight](#)

[FA\\_Pen\\_Shadow](#) [FA\\_Pen\\_Text](#)

### 1.15 FC\_Display / FA\_Pen\_Text

NAME

FA\_Pen\_Text -- (00.00) [I.G], LONG

FUNCTION

This attribute returns the color register used as Text.

It can also be supplied when creating a Display object or a new scheme (using [FM\\_Display\\_CreateScheme](#)) to alter scheme building. See [FM\\_CreatePen](#) to know about pen specifications.

SEE ALSO

[FA\\_Display\\_Scheme](#)

[FA\\_Pen\\_Dark](#) [FA\\_Pen\\_Fill](#)

[FA\\_Pen\\_HalfDark](#) [FA\\_Pen\\_HalfShadow](#)

[FA\\_Pen\\_HalfShine](#) [FA\\_Pen\\_Highlight](#)

[FA\\_Pen\\_Shadow](#) [FA\\_Pen\\_Shine](#)

---

## 1.16 FC\_Display / FA\_Screen

NAME

FA\_Screen -- (00.00) [I.G], PTR TO Screen

FUNCTION

This attribute returns a pointer to the display context's screen.

This attribute can also be supplied at creation time to specify which screen should be used as display context. By default the default public screen is used.

## 1.17 FC\_Display / FA\_Screen\_Depth

NAME

FA\_Screen\_Depth -- (00.00) [..G], LONG

FUNCTION

Current display context's depth.

SEE ALSO

[FA\\_Screen](#)

## 1.18 FC\_Display / FeelinPen

STRUCTURE FeelinPen,FeelinNode\_SIZE ULONG fpen\_R ULONG fpen\_G ULONG fpen\_B ULONG fpen\_Reg UWORD  
fpen\_Flags LABEL FeelinPen\_SIZE

```
struct FeelinPen { struct FeelinNode fpen_Node ULONG fpen_R ULONG fpen_G ULONG fpen_B ULONG fpen_Reg UWORD  
fpen_Flags }
```

OBJECT feelinPen OF feelinNode r,g,b, reg, flags:INT ENDOBJECT

---