

# **FC\_ImageDisplay**

Olivier Laviale 2002

**COLLABORATORS**

	<i>TITLE :</i> FC_ImageDisplay		
<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_ImageDisplay</b>	<b>1</b>
1.1	Feelin : FC_ImageDisplay . . . . .	1
1.2	FC_ImageDisplay / FM_ImageDisplay_Cleanup . . . . .	1
1.3	FC_ImageDisplay / FM_ImageDisplay_Compare . . . . .	2
1.4	FC_ImageDisplay / FM_ImageDisplay_Draw . . . . .	2
1.5	FC_ImageDisplay / FM_ImageDisplay_Setup . . . . .	2
1.6	FC_ImageDisplay / FA_ImageDisplay . . . . .	3
1.7	FC_ImageDisplay / FA_ImageDisplay_ . . . . .	3
1.8	FC_ImageDisplay / FA_ImageDisplay_ . . . . .	3

---

## Chapter 1

# FC\_ImageDisplay

### 1.1 Feelin : FC\_ImageDisplay

FC\_ImageDisplay

ImageDisplay class is used to manage images, as TextDisplay is dedicated to text and FrameDisplay dedicated to frames.

An image may be one of the predefined Feelin patterns, a color, a brush (loaded with DataTypes), a bitmap (loaded with DataTypes), a function of your own... This is completely transparent to the user of the class as everything is handled by the class itself. This class is used by FC\_FrameDisplay and FC\_Image to manage their backgrounds / images.

ImageDisplay uses FA\_Image and FA\_AltImage attributes to create the image. They may describe different image type e.g. a color and a bitmap everything is handled by the class itself.

Brushes and BitMaps are loaded once and are shared by each object using them, updating their data if needed (different screen or pens), saving you a lot of memory (especially for bitmaps).

Methods

[FM\\_ImageDisplay\\_Setup](#) [FM\\_ImageDisplay\\_Cleanup](#)

[FM\\_ImageDisplay\\_Compare](#)

[FM\\_ImageDisplay\\_Draw](#)

Attribute

[FA\\_ImageDisplay](#)

### 1.2 FC\_ImageDisplay / FM\_ImageDisplay\_Cleanup

NAME

FM\_ImageDisplay\_Cleanup -- (00.00)

SYNOPSIS

F\_Do(obj,FM\_ImageDisplay\_Cleanup,struct FS\_Render \*psRender)

FUNCTION

This method cause the ImageDisplay object to free resources used by the image. Colors, Bitmaps... will be freed.

INPUTS

psRender is a pointer to a FS\_Render structure. This structure must be provided as it holds critical information such as the RastPort, the DisplayContext object... If psRender is NULL or one of its critical element, then nothing is done.

SEE ALSO

[FM\\_ImageDisplay\\_Draw](#) [FM\\_ImageDisplay\\_Setup](#)

---

### 1.3 FC\_ImageDisplay / FM\_ImageDisplay\_Compare

NAME

FM\_ImageDisplay\_Compare -- (00.00)

SYNOPSIS

F\_Do(obj,FM\_ImageDisplay\_Compare,APTR paImage)

FUNCTION

This method is used to compare two ImageDisplay objects.

It's the only way you have to know if two ImageDisplay objects represent the same color, raster, bitmap...

INPUTS

paImage is a pointer to an ImageDisplay object to be compared.

RESULT

If the images handled by the objects are the same (same color, raster, bitmap...) the method will return TRUE.

### 1.4 FC\_ImageDisplay / FM\_ImageDisplay\_Draw

NAME

FM\_ImageDisplay\_Draw -- (00.00)

SYNOPSIS

F\_Do(obj,FM\_ImageDisplay\_Draw,struct FS\_Rect \*psRect,struct FS\_Render \*psRender)

FUNCTION

This method is used to draw the ImageDisplay object.

INPUTS

psRect is a pointer to a FS\_Rect structure describing the rectangle to draw in. If (Rect -> x2 < Rect -> x1) or (Rect -> y2 < Rect -> y1) then nothing is drawn.

psRender is a pointer to a FS\_Render structure. This structure must be provided as it holds critical information such as the RPort, the DisplayContext object... If psRender is NULL, or one of its critical element, then nothing is drawn.

SEE ALSO

[FM\\_ImageDisplay\\_Setup](#)

FA\_Image\_State

### 1.5 FC\_ImageDisplay / FM\_ImageDisplay\_Setup

NAME

FM\_ImageDisplay\_Setup -- (00.00)

SYNOPSIS

F\_Do(obj,FM\_ImageDisplay\_Render,struct FS\_Render \*psRender)

FUNCTION

This method cause the ImageDisplay object to render (setup). Colors will be allocated, BitMap remaped, Brushes colored...

INPUTS

---

psRender is a pointer to a FS\_Render structure. This structure must be provided as it holds critical information such as the RastPort, the DisplayContext object... If psRender is NULL or one of its critical element then nothing is done.

SEE ALSO

[FM\\_ImageDisplay\\_Cleanup](#) [FM\\_ImageDisplay\\_Draw](#)

[FA\\_Image\\_State](#)

## 1.6 FC\_ImageDisplay / FA\_ImageDisplay

NAME

FA\_ImageDisplay -- (00.00) [..G], APTR

FUNCTION

This is a general attribute that may be used by other classes. This attribute has no effect on an ImageDisplay object.

## 1.7 FC\_ImageDisplay / FA\_ImageDisplay\_

NAME

FA\_ImageDisplay\_ -- (00.00) [],

FUNCTION

SEE ALSO

## 1.8 FC\_ImageDisplay / FA\_ImageDisplay\_

NAME

FA\_ImageDisplay\_ -- (00.00) [],

FUNCTION

SEE ALSO

---