

FC_Radio

Olivier LAVIALE 2004

COLLABORATORS

	<i>TITLE :</i> FC_Radio		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Olivier LAVIALE 2004	January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	FC_Radio	1
1.1	Feelin : FC_Radio	1
1.2	FC_Radio / FA_Radio_Active	1
1.3	FC_Radio / FA_Radio_Entries	2
1.4	FC_Radio / FA_Radio_PreParse	2
1.5	FC_Radio / FM_Dispose	2
1.6	FC_Radio / FM_New	2

Chapter 1

FC_Radio

1.1 Feelin : FC_Radio

FC_Radio

IDs: Dynamic Super: FC_Group Include: <libraries/feelin.h>

This class generates radio button gadgets. They do the same job as FC_Cycle objects and eat up more window space, maybe that's the reason why so few of them can be found in existing applications.

CHANGES

[FM_New](#) [FM_Dispose](#)

ATTRIBUTES

[FA_Radio_Active](#) [FA_Radio_Entries](#)

[FA_Radio_PreParse](#)

PREFERENCES ITEMS

[FP_Radio_HSpacing](#) [FP_Radio_VSpacing](#)

[FP_Radio_HPreParse](#) [FP_Radio_VPreParse](#)

1.2 FC_Radio / FA_Radio_Active

NAME

[FA_Radio_Active](#) -- (01.00) [ISG], ULONG

FUNCTION

This attribute defines the number of the active entry in the radio gadgets. Valid range is from 0 for the first entry to NumEntries-1 for the last.

Setting [FA_Radio_Active](#) causes the gadget to be updated. On the other hand, when the user plays around with the gadget, [FA_Radio_Active](#) will always reflect the current state.

EXAMPLE

```
F_Set(radioobj,(ULONG) "FA_Radio_Active",3);
```

SEE ALSO

[FA_Radio_Entries](#)

1.3 FC_Radio / FA_Radio_Entries

NAME

FA_Radio_Entries -- (01.00) [I.], STRPTR *

FUNCTION

With this attribute you can define what entries shall be displayed in your radio gadget. You must supply a pointer to a string array, containing one entry for each item and terminated with a NULL.

Remember that radio gadget entries may contain any text formatting code such as bold, italic or underlined characters.

EXAMPLE

```
static STRPTR tbl_Positions[] = { "left","center","right", NULL };
```

```
Obj = RadioObject, "FA_Radio_Entries", tbl_Positions, End;
```

SEE ALSO

[FA_Radio_Active](#) [FA_Text](#)

1.4 FC_Radio / FA_Radio_PreParse

NAME

FA_Radio_PreParse -- (01.00) [I.], STRPTR

FUNCTION

Use this attribute to modify the preparse of all FC_Text object created by the radio gadget. This is useful to center text in vertical mode.

SEE ALSO

[FA_Horizontal](#) [FA_Text_PreParse](#)

[FP_Radio_HPreParse](#) [FP_Radio_VPreParse](#)

1.5 FC_Radio / FM_Dispose

NAME

FM_Dispose

CHANGE

Dispose the array used to remember objects and to handle notifies.

SEE ALSO

[FM_New](#) [FC_Object](#) / [FM_Dispose](#)

1.6 FC_Radio / FM_New

NAME

FM_New

CHANGE

This method is changed to create FC_Image objects and FC_Text objects that will be used to display radio buttons and their label. An array containing pointers to these objects is also created. This array will be used to handle notifications.

The object is created before FC_Image and FC_Text objects. These objects are added dynamically to the FC_Group object's family. The FM_New method is passed to its superclass with the following attributes :

FA_ChainToCycle, FALSE, FA_Group_HSpacing, "FP_Radio_HSpacing", FA_Group_VSpacing, "FP_Radio_VSpacing",
FP_Radio_HSpacing and FP_Radio_VSpacing simply replace FP_Group_HSpacing and FP_Group_VSpacing.

If **FA_Radio_Entries** is NULL, a debug message is logged with level FV_ERLV_DEV.

SEE ALSO

FA_Radio_PreParse FC_Object / FM_Dispose
