FC_Preference

Olivier LAVIALE 2004

COLLABORATORS

	TITLE : FC_Preference	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Olivier LAVIALE 2004	January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FC_Preference

1.1 Feelin : FC_Preference

FC_Preference

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

Feelin is not only a powerful object oriented system with a beautifull GUI but it is also very configurable. The programmer only specifies very few things about the position of particular gadgets, what actually is displayed on screen depends on the users preferences setting.

An instance of this class is used by each application to handle user preferences. The user can define global settings and particuliar (local) settings for each application (relative to the global one).

Since Feelin classes use Dynamic IDs, preference items are not referenced by integer but strings. For example the background setting of the FC_Slider object is referenced as "FP_Slider_Back", this string follows Dynamic rules (no ID are generated thought). Complex mecanims are used to store preference items. As you can imagine strings are more complex to use than IDs, this is why a FC_Dataspace object is not used for this purpose.

This class uses FC_DOSNotify objects to be aware of file changes.

METHODS

FM_Preference_Find FM_Preference_Add

FM_Preference_AddLong FM_Preference_AddString

FM_Preference_Remove FM_Preference_Clear

FM_Preference_Read FM_Preference_Write

FM_Preference_Resolve FM_Preference_ResolveInt

FM_Preference_Update

ATTRIBUTES

FA_Preference_Name FA_Preference_Application

1.2 FC_Preference / FM_Preference_Add

NAME

FM_Preference_Add (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Add,STRPTR Name,APTR Data,ULONG Size);

FUNCTION

Add a new preference item to the object. If a preference item with the same name already exists, it will be replaced with the new one.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it
If the string doesn't follow these rules the item won't be added.

Data - Pointer to the data you want to add.

Size - Size of the Data.

SEE ALSO

FM_Preference_AddLong FM_Preference_AddString

FM_Preference_Find FM_Preference_Remove

1.3 FC_Preference / FM_Preference_AddLong

NAME

FM_Preference_AddLong (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_AddLong,STRPTR Name,ULONG Value);

FUNCTION

Add a new preference item to the object. If a preference item with the same name already exists, it will be replaced with the new one.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it
If the string doesn't follow these rules the item won't be added.

Value - Value to add.

NOTE

This method is a shortcut to F_Do(Obj,FM_Preference_Add,Name,&Value,4)

SEE ALSO

FM_Preference_Add FM_Preference_Find

FM_Preference_Remove

1.4 FC_Preference / FM_Preference_AddString

NAME

FM_Preference_AddString (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_AddString,STRPTR Name,STRPTR String);

FUNCTION

Add a new preference item to the object. If a preference item with the same name already exists, it will be replaced with the new one.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it
If the string doesn't follow these rules the item won't be added.

String - String to add.

NOTE

This method is a shortcut to F_Do(Obj,FM_Preference_Add,Name,String,F_StrLen(String) + 1)

SEE ALSO

FM_Preference_Add FM_Preference_Find

FM_Preference_Remove

1.5 FC_Preference / FM_Preference_Clear

NAME

FM_Preference_Clear (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Clear);

FUNCTION

This method clears the contents of the object. Depending on the state of the memory pool that the object uses, this may or may not result in more free memory.

RESULT

All entries will be removed from the object. The return value of this method is currently undefined.

SEE ALSO

FM_Preference_Add FM_Preference_Remove

1.6 FC_Preference / FM_Preference_Find

NAME

FM_Preference_Find (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Find,STRPTR Name);

FUNCTION

This method will try to find a preference item with the corresponding Name. If it succeed, result will be a pointer to the item's data.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it

SEE ALSO

FM_Preference_Resolve FM_Preference_ResolveInt

1.7 FC_Preference / FM_Preference_Read

NAME

FM_Preference_Read (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Read);

FUNCTION

Clear the object and add the contents of an IFF preference file to the object. Only preference files created with FM_Preference_Write can be read by the method.

SEE ALSO

FM_Preference_Add FM_Preference_Remove

FA_Preference_Name

1.8 FC_Preference / FM_Preference_Remove

NAME

FM_Preference_Remove (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Remove,STRPTR Name);

FUNCTION

Remove a preference item from the object.

INPUTS

Name - name of the preference item. This string must follow Dynamics rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it

SEE ALSO

FM_Preference_Add FM_Preference_Find

1.9 FC_Preference / FM_Preference_Resolve

NAME

FM_Preference_Resolve (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Resolve,STRPTR Name,APTR Default);

FUNCTION

Search through the entries of the object for the given preference item. If an item's name match it, a pointer to the item is returned. If no preference item is found the method is passed to the object managing global preferences. If the item doesn't exists either in local settings nor in global settings, Default is returned.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it
If Name is not valid it is returned to allow easy preference overridings.

Default - This value is returned if the preference item doesn't exists either in local settings nor in global settings.

EXAMPLE

F_METHOD(APTR,mNew} { struct LocalObjectData *LOD = F_LOD(Class,Obj); struct TagItem *Tags = Msg, item;

...

LOD -> p_Back = (ULONG) "FP_String_Background"; // User setting LOD -> p_BlinkSpeed = (ULONG) "FP_String_BlinkSpeed"; // User setting

...

while (F_DynamicNTI(&Tags,&item,Class)) switch (item.ti_Tag) { /* The programmer may override user settings */ ... case FA_String_Background: LOD -> p_Back = item.ti_Data; break; case FA_String_BlinkSpeed: LOD -> p_BlinkSpeed = item.ti_Data; break; ... }

... }

F_METHOD(ULONG,mSetup} { struct LocalObjectData *LOD = F_LOD(Class,Obj);

...

back = F_Do(_app(Obj),FM_Application_Resolve,LOD -> p_Back,"0:3"); speed = F_Do(_app(Obj),FM_Application_ResolveInt,LOD -> p_BlinkSpeed,0);

... }

NOTE

As you can see in the example above, the FM_Application_Resolve method is actually passed to this method as is.

SEE ALSO

FM_Preference_Find FM_Preference_ResolveInt

1.10 FC_Preference / FM_Preference_ResolveInt

NAME

FM_Preference_ResolveInt (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_ResolveInt,STRPTR Name,ULONG Default);

FUNCTION

Search through the entries of the object for the given preference item. If an item's name match it, the value of the item is returned. If no preference item is found the method is passed to the object managing global preferences. If the item doesn't exists either in local settings nor in global settings, Default is returned.

INPUTS

Name - name of the preference item. This string must follows Dynamic rules e.g. "FP_Slider_Back". <identifier>_<collectionname>_<it
If Name is not valid it is returned to allow easy preference overridings.

Default - This value is returned if the preference item doesn't exists either in local settings nor in global settings.

SEE ALSO

FM_Preference_Find FM_Preference_Resolve

1.11 FC_Preference / FM_Preference_Update

NAME

FM_Preference_Update (01.00)

SYNOPSIS

[PRIVATE]

FUNCTION

This method is currently private.

SEE ALSO

FA_Preference_Application

1.12 FC_Preference / FM_Preference_Write

NAME

FM_Preference_Write (01.00)

SYNOPSIS

F_Do(Obj,FM_Preference_Write);

FUNCTION

Writes the contents of the object to an IFF file defined by FA_Preference_Name .

Applications are automatically updated if their local settings changed. All applications are updated if global setting changed.

SEE ALSO

FA_Preference_Application

1.13 FC_Preference / FA_Preference_Name

NAME

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

FUNCTION

Name of the preference file. All disk operations will affect the file.

INPUTS

Pointer to a string defining the name of the preference file. e.g. "ENVARC:Feelin/MyProg.gui".

SEE ALSO

FM_Preference_Read FM_Preference_Write

1.14 FC_Preference / FA_Preference_Application

NAME

FA_Preference_Application -- (01.00) [I..], APTR

FUNCTION

If changes happen to local or global settings the application defined with this attribute is updated.

Using this attribute, you don't have to care about FA_Preference_Name, the name of the application will be used instead. e.g. if you define the FA_Application_Base attribute of your application to "LOLOVE", preferences will be read from "ENV:Feelin/LOLOVE.g and written to "ENVARC:Feelin/LOLOVE.gui"

INPUTS

Pointer to a FC_Application object.

SEE ALSO

FA_Preference_Name