FC_Numeric

Olivier LAVIALE 2004

COLLABORATORS			
	TITLE : FC_Numeric		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Olivier LAVIALE 2004	January 13, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FC_Numeric

1.1 Feelin / FC_Numeric

FC_Numeric

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

This class is the base class for everything that deals with the input (and display) of integer numbers. The class itself does not feature any GUI elements, it just offers some basic attributes and methods. Creating direct instances of this class usually doesn't make any sense. Instead, use one of the included subclasses like FC_Slider, FC_Gauge... to select the type of gadget you need.

FC_Numeric and the supplied subclasses communicate with a set of methods. By writing subclasses which override some of them, you can change the behaviour of all sliders to fit your requirements. You could e.g. enhance the builtin value formatting code which is limited to simple printf-style strings by replacing the FM_Numeric_Stringify method with something more complicated. Or you turn your sliders to logarythmic scales by replacing FM_Numeric_ValueToScale and FM_Numeric_ScaleToValue.

If really none of the supplied subclasses of FC_Numeric suits your requirements, you may of course write custom classes for numeric data input. If you use FC_Numeric, you won't need to think about the basic stuff like min and max values and formatting.

Keyboard control is handled by FC_Numeric automatically, subclasses will not have to care about it.

METHODS

FM_Numeric_Decrease FM_Numeric_Increase

FM_Numeric_Reset FM_Numeric_Stringify

ATTRIBUTES

FA_Numeric_ComputeHook FA_Numeric_Default

FA_Numeric_Format FA_Numeric_Max

FA_Numeric_Min FA_Numeric_Step

FA_Numeric_Value

1.2 FC_Numeric / FM_Numeric_Decrease

NAME

FM_Numeric_Decrease -- (00.00)

SYNOPSIS

F_Do(Obj,FM_Numeric_Decrease,LONG Value);

FUNCTION

Decrease the numeric value of the object.

RESULT

The new value of FA_Numeric_Value .

SEE ALSO

FM_Numeric_Increase FA_Numeric_Step

FA_Numeric_Value

1.3 FC_Numeric / FM_Numeric_Increase

NAME

FM_Numeric_Increase -- (00.00)

SYNOPSIS

F_Do(Obj,FM_Numeric_Increase,LONG Value);

FUNCTION

Increase the numeric value of the object.

RESULT

The new value of FA_Numeric_Value .

SEE ALSO

FM_Numeric_Decrease FA_Numeric_Step

FA_Numeric_Value

1.4 FC_Numeric / FM_Numeric_Reset

NAME

FM_Numeric_Reset -- (00.00)

SYNOPSIS

F_Do(Obj,FM_Numeric_Reset);

FUNCTION

Set the numeric value of the object to its default.

RESULT

The new value of FA_Numeric_Value .

SEE ALSO

FA_Numeric_Default FA_Numeric_Value

1.5 FC_Numeric / FM_Numeric_Stringify

NAME

FM_Numeric_Stringify -- (00.00)

SYNOPSIS

F_Do(Obj,FM_Numeric_Stringify,LONG Value)

FUNCTION

Call this method in your subclass whenever you want to translate a value into a string. A pointer to a string buffer is returned.

Only implementors of custom slider classes will need this method. Sending it from applications doesn't make any sense.

SEE ALSO

FA_Numeric_Format

1.6 FC_Numeric / FA_Numeric_ComputeHook

NAME

FA_Numeric_ComputeHook -- (02.00) [IS.], struct Hook *

FUNCTION

The hook defined with this attribute will be called each time FA_Numeric_Value is modified, the result of the call is set instead of the value specified by FA_Numeric_Value .

1.7 FC_Numeric / FA_Numeric_Default

NAME

FA_Numeric_Default -- (00.00) [ISG], LONG

FUNCTION

Adjust the default value for a numeric input/display gadget. When the object receives a FM_Numeric_Reset method, it sets its value to the one given here.

Each type of object can have a default value to which the user can always return immediately by some action depending on the implementation of the subclass. FC_Slider e.g. resets to defaults after a double click in the knob area.

The default value can also be reached by pressing the toggle key (usually SPACE) on an active numeric gadget.

FA_Numeric_Default defaults to 0.

SEE ALSO

FA_Numeric_Max FA_Numeric_Min

FA_Numeric_Value

1.8 FC_Numeric / FA_Numeric_Format

NAME

FA_Numeric_Format -- (00.00) [ISG], STRPTR

FUNCTION

Printf-style string to describe the format of the numeric display.

Whenever a subclass of FC_Numeric thinks its time to render a new value, it doesn't simply write it to a string but instead calls FM_Numeric_Stringify . This method looks for the specified FA_Numeric_Format in its data structures and fills a string with the current value. In detail, things work like this:

- Some slider object (e.g. a knob) receives a FM_Draw method.

- The FM_Draw implementation of the knob object reads the current value of the Numeric class and calls FM_Numeric_Stringify with this value.

- FM_Numeric_Stringify of FC_Numeric reads the current format and RawDoFmt()s the given value to a buffer. The buffer is returned the caller.

- After all this stuff, the FM_Draw implementation receives a nice string as result code and finally puts it somewhere into the window.

All this method stuff might sound a bit crazy, but in fact its quite powerful. If you write a subclass of any of Feelin's slider classes which simply replaces FM_Numeric_Stringify with your own code, you can create any string you like for display in these sliders. You might e.g. want to display a nice formatted time string (hh:mm:ss) in a slider knob which adjusts a number of seconds. Or you need to adjust a baudrate from a hand of predefined values. Just overrided FM_Numeric_Stringify and you have the choice how the slider value translates into a string.

If you don't override FM_Numeric_Stringify , the method reaches FC_Numeric which simply does a sprintf() with the defined FA_Numeric_Format.

Note well: The maximum length of the result string for FA_Numeric_Format is limited to 64 characters. If you need more, you *must* override the method.

FA_Numeric_Format defaults to "%ld".

SEE ALSO

FM_Numeric_Stringify FA_Numeric_Max

FA_Numeric_Min FA_Numeric_Value

1.9 FC_Numeric / FA_Numeric_Max

NAME

FA_Numeric_Max -- (00.00) [ISG], LONG

FUNCTION

Adjust the maximum value for a numeric input/display gadget.

FC_Numeric will automatically clip its value to make it fit between FA_Numeric_Min and FA_Numeric_Max. Also, minimum and maximum values are used for several internal calculations such as the maximum space required to display a numeric value.

All values in numeric class are treated as signed longwords, so that's the limit for all tags.

FA_Numeric_Max defaults to 100.

SEE ALSO

FA_Numeric_Default FA_Numeric_Max

FA_Numeric_Min FA_Numeric_Step

FA_Numeric_Value

1.10 FC_Numeric / FA_Numeric_Min

NAME

FA_Numeric_Min -- (00.00) [ISG], LONG

FUNCTION

Adjust the minimum value for a numeric input/display gadget.

FC_Numeric will automatically clip its value to make it fit between FA_Numeric_Min and FA_Numeric_Max . Also, minimum and maximum values are used for several internal calculations such as the maximum space required to display a numeric value.

All values in numeric class are treated as signed longwords, so that's the limit for all tags.

FA_Numeric_Min defaults to 0.

SEE ALSO

FA_Numeric_Default FA_Numeric_Max

FA_Numeric_Min FA_Numeric_Step

FA_Numeric_Value

1.11 FC_Numeric / FA_Numeric_Step

NAME

FA_Numeric_Step -- (00.00) [ISG], LONG

FUNCTION

This value is used when the user wants to quickly increment / decrement the numeric value (e.g. holding FV_KEY_STEPxxx while pressing cursor keys).

SEE ALSO

FA_Numeric_Value

1.12 FC_Numeric / FA_Numeric_Value

NAME

FA_Numeric_Value -- (00.00) [ISG], LONG

FUNCTION

Adjust the current value for a numeric input/display gadget. FC_Numeric will automatically clip this value to make it fit between FA_Numeric_Min and FA_Numeric_Max .

Whenever a new value is set, the object's owner receices a FM_Draw method to get a chance to update its display.

The attribute is imported on FM_Import and exported on FM_Export.

SEE ALSO

FM_Numeric_Stringify FA_Numeric_Default

FA_Numeric_Max FA_Numeric_Min

FA_Numeric_Step