NAME

cdk_fscale − curses scale widget (type float).

SYNOPSIS

cc [*flag* . . .] *file* . . . **−lcdk** [*library* . . .]

#include <cdk.h>

float activateCDKFScale (
CDKFSCALE ****scale*,**
chtype ****actions*);**

void destroyCDKFScale (
CDKFSCALE ****scale*);**

void drawCDKFScale (
CDKFSCALE ****scale*,**
boolean *box***);**

void eraseCDKFScale (
CDKFSCALE ****scale*);**

boolean getCDKFScaleBox (
CDKFSCALE ****scale*);**

int getCDKFScaleDigits (
CDKFSCALE ****scale*);**

float getCDKFScaleHighValue (
CDKFSCALE ****scale*);**

float getCDKFScaleLowValue (
CDKFSCALE ****scale*);**

float getCDKFScaleValue (
CDKFSCALE ****scale*);**

int injectCDKFScale (
CDKFSCALE ****scale*,**
chtype *input***);**

void moveCDKFScale (
CDKFSCALE ****scale*,**
int *xpos***,**
int *ypos***,**
boolean *relative***,**
boolean *refresh***);**

CDKFSCALE *newCDKFScale (
CDKSCREEN ****cdkscreen*,**
int *xpos***,**
int *ypos***,**
char ****title*,**
char ****label*,**
chtype *fieldAttribute***,**
int *fieldWidth***,**
float *currentValue***,**
float *lowValue***,**
float *highValue***,**
float *increment***,**
float *fastIncrement***,**
int *digits***,**
boolean *box***,**
boolean *shadow***);**

void positionCDKFScale (
CDKFSCALE ****scale*);**

void setCDKFScale (
CDKFSCALE ****scale*,**
float *lowValue***,**
float *highValue***,**
float *currentValue***,**
boolean *box***);**

void setCDKFScaleBackgroundAttrib (
CDKFSCALE ****scale*,**
chtype *attribute***);**

void setCDKFScaleBackgroundColor (
CDKFSCALE ****scale*,**
char * *color***);**

void setCDKFScaleBox (
CDKFSCALE ****scale*,**
boolean *box***);**

void setCDKFScaleBoxAttribute (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleDigits (
CDKFSCALE ****scale*,**
int *digits***);**

void setCDKFScaleHorizontalChar (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleLLChar (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleLRChar (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleLowHigh (
CDKFSCALE ****scale*,**
float *low***,**
float *high***);**

void setCDKFScalePostProcess (
CDKFSCALE ****scale*,**
PROCESSFN *callback***,**
void * *data***);**

void setCDKFScalePreProcess (
CDKFSCALE ****scale*,**
PROCESSFN *callback***,**
void * *data***);**

void setCDKFScaleULChar (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleURChar (
CDKFSCALE ****scale*,**
chtype *character***);**

void setCDKFScaleValue (
CDKFSCALE ****scale*,**
float *value***);**

void setCDKFScaleVerticalChar (
CDKFSCALE ****scale*,**
chtype *character***);**

DESCRIPTION

The Cdk scale widget creates a scale box with a label and a scale field. The following functions create or manipulate the Cdk scale box widget.

AVAILABLE FUNCTIONS

activateCDKFScale

activates the scale widget and lets the user interact with the widget. The parameter **scale** is a pointer to a non-NULL scale widget. If the **actions** parameter is passed with a non-NULL value, the characters in the array will be injected into the widget. To activate the widget interactively pass in a *NULL* pointer for **actions**. If the character entered into this widget is *RETURN* or *TAB* then this function will return a value from the low value to the high value. It will also set the widget data *exitType* to *vNORMAL*. If the character entered into this widget was *ESCAPE* then the widget will return the unknownFloat value (see the cdk_objs.h header file). The widget data *exitType* will be set to *vESCAPE_HIT*.

destroyCDKFScale

removes the widget from the screen and frees memory the object used.

drawCDKFScale

draws the scale widget on the screen. If the **box** parameter is true, the widget is drawn with a box.

eraseCDKFScale

removes the widget from the screen. This does *NOT* destroy the widget.

getCDKFScaleBox

returns whether the widget will be drawn with a box around it.

getCDKFScaleDigits

returns the number of digits shown after the decimal point for the box value.

getCDKFScaleHighValue

returns the high value of the scale widget.

getCDKFScaleLowValue

returns the low value of the scale widget.

getCDKFScaleValue

returns the current value of the widget.

injectCDKFScale

injects a single character into the widget. The parameter **scale** is a pointer to a non-NULL scale widget. The parameter **character** is the character to inject into the widget. The return value and side-effect (setting the widget data *exitType*) depend upon the injected character:

RETURN or *TAB*

the function returns a value ranging from the scale's low value to the scale's high value. The widget data *exitType* is set to *vNORMAL*.

ESCAPE

the function returns the unknownFloat value (see the cdk_objs.h header file). The widget data *exitType* is set to *vESCAPE_HIT*.

Otherwise

unless modified by preprocessing, postprocessing or key bindings, the function returns the unknownFloat value (see the cdk_objs.h header file). The widget data *exitType* is set to *vEARLY_EXIT*.

moveCDKFScale

moves the given widget to the given position. The parameters **xpos** and **ypos** are the new position of the widget. The parameter **xpos** may be an integer or one of the pre-defined values *TOP*, *BOTTOM*, and *CENTER*. The parameter **ypos** may be an integer or one of the pre-defined values *LEFT*, *RIGHT*, and *CENTER*. The parameter **relative** states whether the **xpos**/**ypos** pair is a relative move or an absolute move. For example, if **xpos** = 1 and **ypos** = 2 and **relative** = **TRUE**,

then the widget would move one row down and two columns right. If the value of **relative** was **FALSE** then the widget would move to the position (1,2). Do not use the values *TOP*, *BOTTOM*, *LEFT*, *RIGHT*, or *CENTER* when **relative** = *TRUE*. (weird things may happen). The final parameter **refresh** is a boolean value which states whether the widget will get refreshed after the move.

newCDKFScale
creates a pointer to a scale widget. Parameters:

screen
is the screen you wish this widget to be placed in.

xpos controls the placement of the object along the horizontal axis. It may be an integer or one of the pre-defined values *LEFT*, *RIGHT*, and *CENTER*.

ypos controls the placement of the object along the vertical axis. It may be an integer or one of the pre-defined values *TOP*, *BOTTOM*, and *CENTER*.

title is the string to display at the top of the widget. The title can be more than one line; just provide a carriage return character at the line break.

label
is the string to display in the label of the scale field.

fieldAttribute
is the attribute of the characters displayed in the field.

fieldWidth
controls the width of the widget. If you provide a value of zero the widget will be created with the full width of the screen. If you provide a negative value, the widget will be created the full width minus the value provided.

currentValue
is the value of the scale field when the widget is activated.

lowValue and

highValue
are the low and high values of the widget respectively.

increment
is the regular increment value

fastIncrement
is the accelerated increment value.

box is true if the widget should be drawn with a box around it.

shadow
turns the shadow on or off around this widget.

If the widget could not be created then a *NULL* pointer is returned.

positionCDKFScale
allows the user to move the widget around the screen via the cursor/keypad keys. See **cdk_position (3)** for key bindings.

setCDKFScale
lets the programmer modify certain elements of an existing scale widget. The parameter names correspond to the same parameter names listed in the *newCDKFScale* function.

setCDKFScaleBackgroundAttrib
sets the background attribute of the widget. The parameter **attribute** is a curses attribute, e.g., A_BOLD.

setCDKFScaleBackgroundColor
sets the background color of the widget. The parameter **color** is in the format of the Cdk format strings. See **cdk_display (3)**.

setCDKFScaleBox
sets whether the widget will be drawn with a box around it.

setCDKFScaleBoxAttribute
sets the attribute of the box.

setCDKFScaleDigits
sets the number of digits shown after the decimal point for the box value.

setCDKFScaleHorizontalChar
sets the horizontal drawing character for the box to the given character.

setCDKFScaleLLChar
sets the lower left hand corner of the widget's box to the given character.

setCDKFScaleLRChar
sets the lower right hand corner of the widget's box to the given character.

setCDKFScaleLowHigh
sets the low and high values of the widget.

setCDKFScalePostProcess
allows the user to have the widget call a function after the key has been applied to the widget. The parameter **function** is the callback function. The parameter **data** points to data passed to the callback function. To learn more about post-processing see *cdk_process (3)*.

setCDKFScalePreProcess
allows the user to have the widget call a function after a key is hit and before the key is applied to the widget. The parameter **function** is the callback function. The parameter **data** points to data passed to the callback function. To learn more about pre-processing see *cdk_process (3)*.

setCDKFScaleULChar
sets the upper left hand corner of the widget's box to the given character.

setCDKFScaleURChar
sets the upper right hand corner of the widget's box to the given character.

setCDKFScaleValue
sets the current value of the widget.

setCDKFScaleVerticalChar
sets the vertical drawing character for the box to the given character.

KEY BINDINGS

When the widget is activated there are several default key bindings which will help the user enter or manipulate the information quickly. The following table outlines the keys and their actions for this widget.

Key	Action
Down Arrow	Decrements the scale by the normal value.
Up Arrow	Increments the scale by the normal value.
u	Increments the scale by the normal value.
Prev Page	Decrements the scale by the accelerated value.
U	Decrements the scale by the accelerated value.
Ctrl-B	Decrements the scale by the accelerated value.
Next Page	Increments the scale by the accelerated value.
Ctrl-F	Increments the scale by the accelerated value.
Home	Sets the scale to the low value.
g	Sets the scale to the low value.
^	Sets the scale to the low value.
End	Sets the scale to the high value.
G	Sets the scale to the high value.
$	Sets the scale to the high value.
Return	Exits the widget and returns the index of the selected value. This also sets the widget data *exitType* to *vNORMAL*.
Tab	Exits the widget and returns the index of the selected value. This also sets the widget data *exitType* to *vNORMAL*.
Escape	Exits the widget and returns the unknownFloat value (see the cdk_objs.h header file). This also sets the widget data *exitType* to *vESCAPE_HIT*.
Ctrl-R	Refreshes the screen.

If the cursor is not pointing to the field's value, the following key bindings apply. You may use the left/right arrows to move the cursor onto the field's value and modify it by typing characters to replace the digits and sign.

Key	Action
Left Arrow	Decrements the scale by the normal value.
Right Arrow	Increments the scale by the normal value.
d	Decrements the scale by the normal value.
D	Increments the scale by the accelerated value.
-	Decrements the scale by the normal value.
+	Increments the scale by the normal value.
0	Sets the scale to the low value.

SEE ALSO

cdk(3), **cdk_binding**(3), **cdk_display**(3), **cdk_position**(3), **cdk_screen**(3)