NAME

cdk_fslider − curses slider widget (type float)

SYNOPSIS

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

#include <cdk.h>

int activateCDKFSlider (
CDKFSLIDER ****slider*,**
chtype ****actions*);**

void destroyCDKFSlider (
CDKFSLIDER ****slider*);**

void drawCDKFSlider (
CDKFSLIDER ****slider*,**
boolean *box***);**

void eraseCDKFSlider (
CDKFSLIDER ****slider*);**

boolean getCDKFSliderBox (
CDKFSLIDER ****slider*);**

int getCDKFSliderDigits (
CDKFSLIDER ****scale*);**

float getCDKFSliderHighValue (
CDKFSLIDER ****slider*);**

float getCDKFSliderLowValue (
CDKFSLIDER ****slider*);**

float getCDKFSliderValue (
CDKFSLIDER ****slider*);**

float injectCDKFSlider (
CDKFSLIDER ****slider*,**
chtype *input***);**

void moveCDKFSlider (
CDKFSLIDER ****slider*,**
int *xpos***,**
int *ypos***,**
boolean *relative***,**
boolean *refresh***);**

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

void positionCDKFSlider (
CDKFSLIDER ****slider*);**

void setCDKFSlider (
CDKFSLIDER ****slider*,**
float *lowValue***,**
float *highValue***,**
float *currentValue***,**
boolean *box***);**

void setCDKFSliderBackgroundAttrib (
CDKFSLIDER ****slider*,**
chtype *attribute***);**

void setCDKFSliderBackgroundColor (
CDKFSLIDER ****slider*,**
char * *color***);**

void setCDKFSliderBox (
CDKFSLIDER ****slider*,**
boolean *boxWidget***);**

void setCDKFSliderBoxAttribute (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderDigits (
CDKFSLIDER ****scale*,**
int *digits***);**

void setCDKFSliderHorizontalChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderLLChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderLRChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderLowHigh (
CDKFSLIDER ****slider*,**
float *lowValue***,**
float *highValue***);**

void setCDKFSliderPostProcess (
CDKFSLIDER ****slider*,**
PROCESSFN *callback***,**
void * *data***);**

void setCDKFSliderPreProcess (
CDKFSLIDER ****slider*,**
PROCESSFN *callback***,**
void * *data***);**

void setCDKFSliderULChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderURChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

void setCDKFSliderValue (
CDKFSLIDER ****slider*,**
float *value***);**

void setCDKFSliderVerticalChar (
CDKFSLIDER ****slider*,**
chtype *character***);**

DESCRIPTION

The Cdk slider widget creates a visual slider box with a label and a slider field. The following are functions which create or manipulate the Cdk slider box widget.

AVAILABLE FUNCTIONS

activateCDKFSlider

activates the slider widget and lets the user interact with the widget. The parameter **slider** is a pointer to a non-NULL slider 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 returns the unknownFloat value (see the cdk_objs.h header file). The widget data *exitType* is set to *vESCAPE_HIT*.

destroyCDKFSlider

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

drawCDKFSlider

draws the slider widget on the screen. If the **box** option is true, the widget is drawn with a box.

eraseCDKFSlider

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

getCDKFSliderBox

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

getCDKFSliderDigits

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

getCDKFSliderHighValue

returns the high value of the slider widget.

getCDKFSliderLowValue

returns the low value of the slider widget.

getCDKFSliderValue

returns the current value of the widget.

injectCDKFSlider

injects a single character into the widget. The parameter **slider** is a pointer to a non-NULL slider 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 slider's low value to the slider'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*.

moveCDKFSlider

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.

newCDKFSlider
function creates a slider widget and returns a pointer to it.. 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 slider field.

fillerCharacter
is the character to be used to draw the slider bar 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 slider 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.

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

setCDKFSlider
lets the programmer modify certain elements of an existing slider widget. The parameter names correspond to the same parameter names listed in the *newCDKFSlider* function.

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

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

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

setCDKFSliderBoxAttribute
function sets the attribute of the box.

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

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

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

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

setCDKFSliderLowHigh
sets the low and high values of the widget.

setCDKFSliderPostProcess
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)*.

setCDKFSliderPreProcess
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)*.

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

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

setCDKFSliderValue
sets the current value of the widget.

setCDKFSliderVerticalChar
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 field by the normal decrement value.
Up Arrow	Increments the field by the normal increment value.
u	Increments the field by the normal increment value.
Prev Page	Decrements the field by the accelerated decrement value.
U	Decrements the field by the accelerated decrement value.
Ctrl-B	Decrements the field by the accelerated decrement value.
Next Page	Increments the field by the accelerated increment value.
D	Increments the field by the accelerated increment value.
Ctrl-F	Increments the field by the accelerated increment value.
Home	Sets the value to the low value.
g	Sets the value to the low value.
End	Sets the value to the high value.
G	Sets the value to the high value.
$	Sets the value to the high value.
Return	Exits the widget and returns the current value. This also sets the widget data *exitType* to *vNORMAL*.
Tab	Exits the widget and returns the current 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)