

cdk_fselect(3)

cdk_fselect(3)

NAME

cdk_fselect – curses file selector widget.

SYNOPSIS

```
cc [ flag ... ] file ... -lcdk [ library ... ]
#include <cdk.h>

char *activateCDKFselect (
    CDKFSELECT *fselect,
    chtype *actions);

void deleteFileCB (
    EObjectType objectType,
    void *object,
    void *clientData);

void destroyCDKFselect (
    CDKFSELECT *fselect);

void drawCDKFselect (
    CDKFSELECT *fselect,
    boolean box);

void eraseCDKFselect (
    CDKFSELECT *fselect);

boolean getCDKFselectBox (
    CDKFSELECT *fselect);

char **getCDKFselectContents (
    CDKFSELECT *widget,
    int *size);

int getCDKFselectCurrentItem (
    CDKFSELECT *widget);

char *getCDKFselectDirAttribute (
    CDKFSELECT *fselect);

char **getCDKFselectDirContents (
    CDKFSELECT *fselect,
    int *count);

char *getCDKFselectDirectory (
    CDKFSELECT *fselect);

char *getCDKFselectFileAttribute (
    CDKFSELECT *fselect);

chtype getCDKFselectFillerChar (
    CDKFSELECT *fselect);

chtype getCDKFselectHighlight (
    CDKFSELECT *fselect);

char *getCDKFselectLinkAttribute (
    CDKFSELECT *fselect);

char *getCDKFselectSocketAttribute (
    CDKFSELECT *fselect);

char *injectCDKFselect (
    CDKFSELECT *fselect,
    chtype input);

void moveCDKFselect (
    CDKFSELECT *fselect,
    int xpos,
    int ypos,
    boolean relative,
```



cdk_fselect(3)

cdk_fselect(3)

```

        boolean refresh);

CDKFSELECT *newCDKFselect (
    CDKSCREEN *cdkscreen,
    int xpos,
    int ypos,
    int height,
    int width,
    char *title,
    char *label,
    chtype fieldAttribute,
    chtype fillerCharacter,
    chtype highlight,
    char *dirAttribute,
    char *fileAttribute,
    char *linkAttribute,
    char *sockAttribute,
    boolean box,
    boolean shadow);

void positionCDKFselect (
    CDKFSELECT *fselect);

void setCDKFselect (
    CDKFSELECT *fselect,
    char *directory,
    chtype fieldAttribute,
    chtype fillerCharacter,
    chtype highlight,
    char *dirAttribute,
    char *fileAttribute,
    char *linkAttribute,
    char *sockAttribute,
    boolean box);

void setCDKFselectBackgroundAttrib (
    CDKFSELECT *fselect,
    chtype attribute);

void setCDKFselectBackgroundColor (
    CDKFSELECT *fselect,
    char *color);

void setCDKFselectBox (
    CDKFSELECT *fselect,
    boolean box);

void setCDKFselectBoxAttribute (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectContents (
    CDKFSELECT *widget,
    char **list,
    int listSize);

void setCDKFselectCurrentItem (
    CDKFSELECT *widget,
    int item);

void setCDKFselectDirAttribute (
    CDKFSELECT *fselect,
    char *attribute);

```



cdk_fselect(3)

cdk_fselect(3)

```

int setCDKFselectDirContents (
    CDKFSELECT *fselect);

int setCDKFselectDirectory (
    CDKFSELECT *fselect,
    char *directory);

void setCDKFselectFileAttribute (
    CDKFSELECT *fselect,
    char *attribute);

void setCDKFselectFillerChar (
    CDKFSELECT *fselect,
    chtype filler);

void setCDKFselectHighlight (
    CDKFSELECT *fselect,
    chtype highlight);

void setCDKFselectHorizontalChar (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectLLChar (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectLRChar (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectLinkAttribute (
    CDKFSELECT *fselect,
    char *attribute);

void setCDKFselectSocketAttribute (
    CDKFSELECT *fselect,
    char *attribute);

void setCDKFselectULChar (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectURChar (
    CDKFSELECT *fselect,
    chtype character);

void setCDKFselectVerticalChar (
    CDKFSELECT *fselect,
    chtype character);

```

DESCRIPTION

The Cdk file selector widget creates a widget which allows a user to interact with the Unix file system. This widget allows the user to traverse the file system and even delete files. (See *NOTES* below for more information)

AVAILABLE FUNCTIONS

activateCDKFselect

activates the file selector widget and lets the user interact with the widget. The parameter **fselect** is a pointer to a non-NULL file selector 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 *char ** representing the information typed into the widget and the widget data *exitType* will be set to *vNORMAL*. If the character entered was *ESCAPE* then the function will return *NULL* pointer and the widget data *exitType* is set to *vESCAPE_HIT*. void deleteFileCB (EObjectType objectType GCC_UNUSED, void *object, void *clientData)



cdk_fselect(3)

cdk_fselect(3)

deleteFileCB

Prompts the user to confirm, and deletes the current item in the file-list.

destroyCDKFselect

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

drawCDKFselect

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

eraseCDKFselect

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

getCDKFselectBox

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

getCDKFselectContents

returns the directory-contents (which are formatted), and sets the number of entries via the *size* parameter.

getCDKFselectCurrentItem

returns the current position in the scroll-widget.

getCDKFselectDirAttribute

returns the attribute that directories are drawn with in the scrolling region of the widget.

getCDKFselectDirContents

returns the list of files and the corresponding count as a side-effect. If no files are found, this function returns zero, otherwise one.

getCDKFselectDirectory

returns the current directory of the widget.

getCDKFselectFileAttribute

returns the attribute that plain files are drawn with in the scrolling region of the widget.

getCDKFselectFillerChar

returns the character being used as the filler character in the entry field portion of the widget.

getCDKFselectHighlight

returns the attribute of the highlight bar of the scrolling list portion of the widget.

getCDKFselectLinkAttribute

returns the attribute that symbolic links are drawn with in the scrolling region of the widget.

getCDKFselectSocketAttribute

returns the attribute that sockets are drawn with in the scrolling region of the widget.

injectCDKFselect

injects a single character into the widget. The parameter **fselect** is a pointer to a non-NULL file selector 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 *char ** representing the information typed into the widget. The widget data *exitType* is set to *vNORMAL*.

ESCAPE

the function returns a *NULL* pointer. The widget data *exitType* is set to *vESCAPE_HIT*.

Otherwise

unless modified by preprocessing, postprocessing or key bindings, the function returns a *NULL* pointer. The widget data *exitType* is set to *vEARLY_EXIT*.

moveCDKFselect

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



cdk_fselect(3)

cdk_fselect(3)

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.

newCDKFselect

creates an fselect widget and returns a pointer to it. Parameters:

The **screen** parameter

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

The parameter **xpos**

controls the placement of the object along the horizontal axis. This parameter 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 value or one of the pre-defined values *TOP*, *BOTTOM*, and *CENTER*.

height and

width

control the height and width of the widget. If you provide a value of zero for either of the height or the width, the widget will be created with the full width and height of the screen. If you provide a negative value, the widget will be created the full height or width minus the value provided.

title is the string which will be displayed 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 text to use as the label of the file selector.

fieldAttribute

is the attribute of the entry field at the top of the widget;

fillerCharacter

is the filler character for the entry field.

highlight

is the highlight attribute on the scrolling list.

dirAttribute,

fileAttribute,

linkAttribute and

sockAttribute

are the attributes of directories, files, links, and sockets found in the directory list.

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.

positionCDKFselect

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

setCDKFselect

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

setCDKFselectBackgroundAttrib

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

setCDKFselectBackgroundColor

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



cdk_fselect(3)

cdk_fselect(3)

setCDKFselectBox

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

setCDKFselectBoxAttribute

sets the attribute of the box.

setCDKFselectContents

sets the contents of the widget.

setCDKFselectCurrentItem

set the current position in the scroll-widget.

setCDKFselectDirAttribute

sets the attribute that directories will be drawn with in the scrolling region of the widget. **setCDKFselectDirContents** This creates a list of the files in the current directory. See **getCDKFselectDirContents**.

setCDKFselectDirectory

sets the current directory of the widget.

setCDKFselectFileAttribute

sets the attribute that plain files will be drawn with in the scrolling region of the widget.

setCDKFselectFillerChar

sets the character to use when drawing the entry field portion of the widget.

setCDKFselectHighlight

sets the attribute of the highlight bar of the scrolling list portion of the widget.

setCDKFselectHorizontalChar

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

setCDKFselectLLChar

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

setCDKFselectLRChar

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

setCDKFselectLinkAttribute

sets the attribute that symbolic links will be drawn with in the scrolling region of the widget.

setCDKFselectSocketAttribute

sets the attribute that sockets will be drawn with in the scrolling region of the widget.

setCDKFselectULChar

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

setCDKFselectURChar

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

setCDKFselectVerticalChar

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. Since this widget is built from both the scrolling list widget and the entry field widget, the key bindings are the same for the respective fields. The extra key bindings are listed below.



cdk_fselect(3)

cdk_fselect(3)

Key	Action
CTRL-^	Displays statistics about the current filename.
Tab	Tries to complete the current filename.
Up Arrow	Scrolls the scrolling list up one line.
Down Arrow	Scrolls the scrolling list down one line.
Page Up	Scrolls the scrolling list up one page.
CTRL-B	Scrolls the scrolling list up one page.
Page Down	Scrolls the scrolling list down one page.
CTRL-F	Scrolls the scrolling list down one page.
Return	Returns the filename in the entry field of the file selector. It also sets the widget data <i>exitType</i> to <i>vNORMAL</i> .
Escape	Exits the widget and returns a <i>NULL</i> pointer. It also sets the widget data <i>exitType</i> to <i>vESCAPE_HIT</i> .

SEE ALSO

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

NOTES

This widget is created from the scrolling list widget and the entry field widget. This is a good example on how to build your own widgets using the base widgets provided in this distribution. In order for you to delete files using the file selector, you must add the following call into your program. (the character X is the character you wish to use to tell the file selector you want to delete a file.)

bindCDKObject (vSCROLL, fselect->filelist, X, deleteFile, fselect);

where **fselect** is a pointer to the object. This was originally turned on, but I thought it better to put the control of file deletion on an 'as-need' basis.

