

cdk_label(3)

cdk_label(3)

NAME

cdk_label – Creates a managed curses label widget.

SYNOPSIS

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

void activateCDKLabel (
    CDKLABEL *label,
    chtype unused);

void destroyCDKLabel (
    CDKLABEL *label);

void drawCDKLabel (
    CDKLABEL *label,
    boolean box);

void eraseCDKLabel (
    CDKLABEL *label);

boolean getCDKLabelBox (
    CDKLABEL *label);

chtyle **getCDKLabelMessage (
    CDKLABEL *label
    int *messageLines);

void moveCDKLabel (
    CDKLABEL *label,
    int xpos,
    int ypos,
    boolean relative,
    boolean refresh);

CDKLABEL *newCDKLabel (
    CDKSCREEN *cdkscreen,
    int xpos,
    int ypos,
    CDK_CONST char **message,
    int messageLines,
    boolean box,
    boolean shadow);

void positionCDKLabel (
    CDKLABEL *label);

void setCDKLabel (
    CDKLABEL *label,
    CDK_CONST char **message,
    int messageLines,
    boolean box);

void setCDKLabelBackgroundAttrib (
    CDKLABEL *label,
    chtype attribute);

void setCDKLabelBackgroundColor (
    CDKLABEL *label,
    const char * color);

void setCDKLabelBox (
    CDKLABEL *label,
    boolean box);
```



cdk_label(3)

cdk_label(3)

```

void setCDKLabelBoxAttribute (
    CDKLABEL *label,
    chtype character);

void setCDKLabelHorizontalChar (
    CDKLABEL *label,
    chtype character);

void setCDKLabelLLChar (
    CDKLABEL *label,
    chtype character);

void setCDKLabelLRChar (
    CDKLABEL *label,
    chtype character);

void setCDKLabelMessage (
    CDKLABEL *label,
    CDK_CONST char **message,
    int messageLines);

void setCDKLabelULChar (
    CDKLABEL *label,
    chtype character);

void setCDKLabelURChar (
    CDKLABEL *label,
    chtype character);

void setCDKLabelVerticalChar (
    CDKLABEL *label,
    chtype character);

char waitCDKLabel (
    CDKLABEL *label,
    char key);

```

DESCRIPTION

The Cdk label widget creates a pop-up label window. The following functions create or manipulate the Cdk label widget.

AVAILABLE FUNCTIONS

activateCDKLabel

obsolete entrypoint which calls **drawCDKLabel**.

destroyCDKLabel

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

drawCDKLabel

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

eraseCDKLabel

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

getCDKLabelBox

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

getCDKLabelMessage

returns the contents of the label widget. The parameter **messageLines** points to the location which will be given the number of message-lines.

moveCDKLabel

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*.



cdk_label(3)

cdk_label(3)

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.

newCDKLabel

creates a label 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*.

message

is the message to be displayed in the window. It may contain any of the standard Cdk display format commands. To learn more about this see *cdk_display (3)*.

rows

is the number of message rows that were passed in the **message** parameter.

box is true if 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.

positionCDKLabel

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

setCDKLabel

allows the user to change the contents of the label widget. The parameters are the same as the **newCDKLabel**.

setCDKLabelBackgroundAttrib

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

setCDKLabelBackgroundColor

sets the background color of the widget. The parameter **color** is in the format of the Cdk format strings. For more information see *cdk_display (3)*.

setCDKLabelBox

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

setCDKLabelBoxAttribute

sets the attribute of the box.

setCDKLabelHorizontalChar

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

setCDKLabelLLChar

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



[cdk_label\(3\)](#)[cdk_label\(3\)](#)**setCDKLabelLRChar**

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

setCDKLabelMessage

This sets the contents of the label widget.

setCDKLabelULChar

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

setCDKLabelURChar

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

setCDKLabelVerticalChar

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

waitCDKLabel

waits for a user to press a key. The **label** parameter is the pointer to a label widget, and **key** is the key to wait for. If no specific key is desired, use *(char)0*.

SEE ALSO

[cdk\(3\)](#), [cdk_binding\(3\)](#), [cdk_display\(3\)](#), [cdk_position\(3\)](#), [cdk_screen\(3\)](#)

