

cdk\_marquee(3)

cdk\_marquee(3)

**NAME**

cdk\_marquee – curses marquee widget

**SYNOPSIS**

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

int activateCDKMarquee (
    CDKMARQUEE *marquee,
    const char *message,
    int delay,
    int repeat,
    boolean box);

void deactivateCDKMarquee (
    CDKMARQUEE *marquee);

void destroyCDKMarquee (
    CDKMARQUEE *marquee);

void drawCDKMarquee (
    CDKMARQUEE *marquee,
    boolean box);

void eraseCDKMarquee (
    CDKMARQUEE *marquee);

boolean getCDKMarqueeBox (
    CDKMENTRY *mentry);

void moveCDKMarquee (
    CDKMARQUEE *marquee,
    int xpos,
    int ypos,
    boolean relative,
    boolean refresh);

CDKMARQUEE *newCDKMarquee (
    CDKSCREEN *cdkscreen,
    int xpos,
    int ypos,
    int fieldWidth,
    boolean box,
    boolean shadow);

void positionCDKMarquee (
    CDKMARQUEE *marquee);

void setCDKMarqueeBackgroundAttrib (
    CDKMARQUEE *marquee,
    chtype attribute);

void setCDKMarqueeBackgroundColor (
    CDKMARQUEE *marquee,
    const char * color);

void setCDKMarqueeBox (
    CDKMENTRY *mentry,
    boolean boxWidget);

void setCDKMarqueeBoxAttribute (
    CDKMARQUEE *marquee,
    chtype character);

void setCDKMarqueeHorizontalChar (
    CDKMARQUEE *marquee,
    chtype character);
```



cdk\_marquee(3)

cdk\_marquee(3)

```

void setCDKMarqueeLLChar (
    CDKMARQUEE *marquee,
    chtype character);

void setCDKMarqueeLRChar (
    CDKMARQUEE *marquee,
    chtype character);

void setCDKMarqueeULChar (
    CDKMARQUEE *marquee,
    chtype character);

void setCDKMarqueeURChar (
    CDKMARQUEE *marquee,
    chtype character);

void setCDKMarqueeVerticalChar (
    CDKMARQUEE *marquee,
    chtype character);

```

**DESCRIPTION**

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

**AVAILABLE FUNCTIONS****activateCDKMarquee**

activates the marquee widget.

- The **marquee** parameter is a pointer to a defined marquee widget.
- The **delay** parameter states how long to wait between movements. This value is highly dependent on each machine the program runs on.
- The **repeat** value tells the marquee widget how many times to display the given message.
- The **box** option draws the widget with or without a box. This function returns -1 if the message passed is *NULL*, 0 otherwise.

**deactivateCDKMarquee**

deactivates the marquee widget.

**destroyCDKMarquee**

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

**drawCDKMarquee**

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

**eraseCDKMarquee**

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

**getCDKMarqueeBox**

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

**moveCDKMarquee**

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).



**cdk\_marquee(3)****cdk\_marquee(3)**

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.

**newCDKMarquee**

creates a marquee 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*.

**width**

controls the width of the field. If you provide a value of zero for the width, the widget uses the width of the screen. If a negative value is provided, then the widget uses the width of the screen less the value provided.

**shadow**

turns the shadow on or off around this widget.

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

**positionCDKMarquee**

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

**setCDKMarqueeBackgroundAttrib**

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

**setCDKMarqueeBackgroundColor**

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)*.

**setCDKMarqueeBox**

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

**setCDKMarqueeBoxAttribute**

sets the attribute of the box.

**setCDKMarqueeHorizontalChar**

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

**setCDKMarqueeLLChar**

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

**setCDKMarqueeLRChar**

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

**setCDKMarqueeULChar**

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

**setCDKMarqueeURChar**

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

**setCDKMarqueeVerticalChar**

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

**SEE ALSO**

**cdk(3)**, **cdk\_binding(3)**, **cdk\_display(3)**, **cdk\_position(3)**, **cdk\_screen(3)**

