

cdk_calendar(3)

cdk_calendar(3)

NAME

cdk_calendar – curses calendar widget.

SYNOPSIS

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

time_t activateCDKCalendar (CDKCALENDAR * calendar ,
                           chtype *actions);

void destroyCDKCalendar (
    CDKCALENDAR *calendar);

void drawCDKCalendar (CDKCALENDAR * calendar ,
                     boolean box);

void eraseCDKCalendar (
    CDKCALENDAR *calendar);

boolean getCDKCalendarBox (
    CDKCALENDAR *calendar);

void getCDKCalendarDate (CDKCALENDAR * calendar ,
                        int *day,
                        int *month,
                        int *year);

chtype getCDKCalendarDayAttribute (
    CDKCALENDAR *calendar);

chtype getCDKCalendarHighlight (
    CDKCALENDAR *calendar);

chtype getCDKCalendarMarker (
    CDKCALENDAR *calendar,
    int day,
    int month,
    int year);

chtype getCDKCalendarMonthAttribute (
    CDKCALENDAR *calendar);

chtype getCDKCalendarYearAttribute (
    CDKCALENDAR *calendar);

time_t injectCDKCalendar (
    CDKCALENDAR *calendar,
    chtype input);

void moveCDKCalendar (
    CDKCALENDAR *calendar,
    int xpos,
    int ypos,
    boolean relative,
    boolean refresh);

CDKCALENDAR *newCDKCalendar (CDKSCREEN * cdkscreen ,
                             int xpos,
                             int ypos,
                             const char *title,
                             int day,
                             int month,
                             int year,
                             chtype dayAttrib,
                             chtype monthAttrib,
                             chtype yearAttrib,
                             chtype highlight,
```



cdk_calendar(3)

cdk_calendar(3)

```
        boolean box,
        boolean shadow);

void positionCDKCalendar (
        CDKCALENDAR *calendar);

void removeCDKCalendarMarker (
        CDKCALENDAR *calendar,
        int day,
        int month,
        int year);

void setCDKCalendar (
        CDKCALENDAR *calendar,
        int day,
        int month,
        int year,
        ctype dayAttrib,
        ctype monthAttrib,
        ctype yearAttrib,
        ctype highlight,
        boolean box);

void setCDKCalendarBackgroundAttrib (
        CDKCALENDAR *calendar,
        ctype attribute);

void setCDKCalendarBackgroundColor (
        CDKCALENDAR *calendar,
        const char * color);

void setCDKCalendarBox (
        CDKCALENDAR *calendar,
        boolean box);

setCDKCalendarBoxAttribute ("
        CDKCALENDAR *calendar,
        ctype ch);

void setCDKCalendarDate (
        CDKCALENDAR *calendar,
        int day,
        int month,
        int year);

void setCDKCalendarDayAttribute (
        CDKCALENDAR *calendar,
        ctype attribute);

void setCDKCalendarDaysNames (
        CDKCALENDAR *calendar,
        const char *days);

void setCDKCalendarHighlight (
        CDKCALENDAR *calendar,
        ctype attribute);

void setCDKCalendarHorizontalChar (
        CDKCALENDAR *calendar,
        ctype ch);

void setCDKCalendarLLChar (
        CDKCALENDAR *calendar,
        ctype ch);
```



cdk_calendar(3)

cdk_calendar(3)

```

void setCDKCalendarLRChar (
    CDKCALENDAR *calendar,
    chtype ch);

void setCDKCalendarMarker (
    CDKCALENDAR *calendar,
    int day,
    int month,
    int year,
    chtype marker);

void setCDKCalendarMonthAttribute (
    CDKCALENDAR *calendar,
    chtype attribute);

void setCDKCalendarMonthsNames (
    CDKCALENDAR *calendar,
    CDK_CONST char **months);

void setCDKCalendarPostProcess (
    CDKCALENDAR *calendar,
    PROCESSFN callback,
    void * data);

void setCDKCalendarPreProcess (
    CDKCALENDAR *calendar,
    PROCESSFN callback,
    void * data);

setCDKCalendarULChar ("
    CDKCALENDAR *calendar,
    chtype ch);

setCDKCalendarURChar ("
    CDKCALENDAR *calendar,
    chtype ch);

setCDKCalendarVerticalChar ("
    CDKCALENDAR *calendar,
    chtype ch);

void setCDKCalendarYearAttribute (
    CDKCALENDAR *calendar,
    chtype attribute);

```

DESCRIPTION

The Cdk calendar widget creates a pop-up calendar. The calendar widget allows the user to traverse through months/years using the cursor keys.

AVAILABLE FUNCTIONS

activateCDKCalendar

activates the calendar widget and lets the user interact with the widget.

- The **calendar** widget is a pointer to a non-NULL calendar 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* then this function will return a type of *time_t*.

- The *time_t* type is used in the functions defined in the *time.h* header file. (see *localtime* or *ctime* for more information).

If the character entered into this widget was *ESCAPE* or *TAB* then this function will return a value of *(time_t)-1* and the widget data *exitType* will be set to *vESCAPE_HIT*.



cdk_calendar(3)

cdk_calendar(3)

destroyCDKCalendar

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

drawCDKCalendar

draws the label widget on the screen.

If the **box** parameter is true, the widget is drawn with a box.

eraseCDKCalendar

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

getCDKCalendarBox

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

getCDKCalendarDate

returns the current date the calendar is displaying.

getCDKCalendarDayAttribute

returns the attribute of the day attribute of the calendar.

getCDKCalendarHighlight

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

getCDKCalendarMarker

returns the marker set on the calendar by **setCDKCalendarMarker**.

getCDKCalendarMonthAttribute

returns the attribute of the month attribute of the calendar.

getCDKCalendarYearAttribute

returns the attribute of the year attribute of the calendar.

injectCDKCalendar

injects a single character into the widget.

- The parameter **calendar** is a pointer to a non-NULL calendar 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 of type *time_t* (see *localtime* or *ctime* for more information).
The widget data *exitType* is set to *vNORMAL*.

ESCAPE

the function returns (time_t)-1. The widget data *exitType* is set to *vESCAPE_HIT*.

Otherwise

unless modified by preprocessing, postprocessing or key bindings, the function returns (time_t)-1. The widget data *exitType* is set to *vEARLY_EXIT*.

moveCDKCalendar

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).



cdk_calendar(3)

cdk_calendar(3)

- The final parameter **refresh** is a boolean value which states whether the widget will get refreshed after the move.

newCDKCalendar

creates a calendar 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 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.

year,

month and

day

set the initial date of the calendar.

yearAttrib,

monthAttrib and

dayAttrib

represent the attributes of the year, month, and day respectively.

highlight

sets the highlight of the currently selected day.

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.

positionCDKCalendar

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

removeCDKCalendarMarker

removes a marker from the calendar.

setCDKCalendar

lets the programmer modify certain elements of an existing calendar widget.

- The **calendar** parameter represents an existing calendar pointer.
- The other parameter names correspond to the same parameter names listed in the **newCDKCalendar** function.

setCDKCalendarBackgroundAttrib

sets the background attribute of the widget.

The parameter **attribute** is a curses attribute, e.g., *A_BOLD*.

setCDKCalendarBackgroundColor

sets the background color of the widget.

The parameter **color** is in the format of the Cdk format strings.

See **cdk_display (3)**.

setCDKCalendarBox

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

setCDKCalendarBoxAttribute

sets the attribute of the box.



cdk_calendar(3)

cdk_calendar(3)

setCDKCalendarDate

sets the calendar to the given date.

If the value of the **day**, **month**, or **year** parameters is -1 then the current day, month, or year is used to set the date.

setCDKCalendarDayAttribute

sets the attribute of the day in the calendar.

setCDKCalendarDaysNames

sets the names of the days of the week.

The parameter is a string listing the 2-character abbreviations for the days.

The default value is

"Su Mo Tu We Th Fr Sa"

"Su" (Sunday) is numbered zero internally, making it by default the first day of the week. Set the **weekBase** member of the widget to select a different day. For example, Monday would be 1, Tuesday 2, etc.

setCDKCalendarHighlight

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

setCDKCalendarHorizontalChar

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

setCDKCalendarLLChar

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

setCDKCalendarLRChar

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

setCDKCalendarMarker

allows the user to set a marker which will be displayed when the month is drawn.

The **marker** parameter is the attribute to use when drawing the marker. If more than one marker is set on a single day, then the day will blink with the original marker attribute.

setCDKCalendarMonthAttribute

sets the attribute of the month in the calendar.

setCDKCalendarMonthsNames

sets the names of the months of the year.

The parameter is a 13-element array (since indexing starts with 1).

The default values are the English month names, e.g., "January".

setCDKCalendarPostProcess

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)*.

setCDKCalendarPreProcess

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)*.

setCDKCalendarULChar

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

setCDKCalendarURChar

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



cdk_calendar(3)

cdk_calendar(3)

setCDKCalendarVerticalChar

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

setCDKCalendarYearAttribute

sets the attribute of the year in the calendar.

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
Left Arrow	Moves the cursor to the previous day.
Right Arrow	Moves the cursor to the next day.
Up Arrow	Moves the cursor to the next week.
Down Arrow	Moves the cursor to the previous week.
t	Sets the calendar to the current date.
T	Sets the calendar to the current date.
n	Advances the calendar one month ahead.
N	Advances the calendar six months ahead.
p	Advances the calendar one month back.
P	Advances the calendar six months back.
-	Advances the calendar one year ahead.
+	Advances the calendar one year back.
Return	Exits the widget and returns a value of <i>time_t</i> which represents the day selected at 1 second after midnight. This also sets the widget data <i>exitType</i> to <i>vNORMAL</i> .
Tab	Exits the widget and returns a value of <i>time_t</i> which represents the day selected at 1 second after midnight. This also sets the widget data <i>exitType</i> to <i>vNORMAL</i> .
Escape	Exits the widget and returns (time_)-1. This also sets the widget data <i>exitType</i> to <i>vESCAPE_HIT</i> .
Ctrl-L	Refreshes the screen.

SEE ALSO

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

