NAME

slk_init, slk_set, slk_wset, slk_refresh, slk_noutrefresh, slk_label, slk_clear, slk_restore, slk_touch, slk_attron, slk_attrset, slk_attroff, slk_attr_on, slk_attr_set, slk_attr_off, slk_attr, slk_color, extended_slk_color - *curses* soft label key routines

SYNOPSIS

#include <curses.h>

int slk_init(int *fmt***);**

int slk_set(int *labnum***, const char ****label***, int** *align***);**
int slk_wset(int *labnum***, const wchar_t ****label***, int** *align***);**

char *slk_label(int *labnum***);**

int slk_refresh(void);
int slk_noutrefresh(void);
int slk_clear(void);
int slk_restore(void);
int slk_touch(void);

int slk_attron(const chtype *attrs***);**
int slk_attroff(const chtype *attrs***);**
int slk_attrset(const chtype *attrs***);**
int slk_attr_on(attr_t *attrs***, void ****opts***);**
int slk_attr_off(const attr_t *attrs***, void ****opts***);**
int slk_attr_set(const attr_t *attrs***, short** *pair***, void****opts***);**
/* extension */
attr_t slk_attr(void);

int slk_color(short *pair***);**
/* extension */
int extended_slk_color(int *pair***);**

DESCRIPTION

These functions manipulate the soft function key labels that some hardware terminals support. For those terminals that do not have soft labels, *curses* takes over the bottom line of **stdscr**, reducing its vertical size and the value of **LINES** by one. By default, *curses* uses eight labels of up to eight characters each.

ncurses furthermore supports a mode comprising twelve labels of up to five characters each, following a convention associated with the IBM PC/AT keyboard. *ncurses* simulates this mode by taking over up to two lines at the bottom of the screen; it does not try to use any hardware support for this mode.

Initialization

slk_init must be called before **initscr** or **newterm**. If **initscr** eventually uses a line from **stdscr** to emulate the soft labels, then *fmt* determines how the labels are arranged on the screen.

0   indicates a 3-2-3 arrangement of the labels.

1   indicates a 4-4 arrangement

2   indicates the PC-like 4-4-4 mode.

3   is again the PC-like 4-4-4 mode, but in addition an index line is generated, helping the user to associate each label with its numbered function key. **LINES** and the vertical size of **stdscr** are further reduced.

Labels

Populate the labels with normal strings (**slk_set**) or wide-character strings (**slk_wset**). Each function takes three parameters.

labnum   is the label number, from **1** to **8** (12 if *fmt* in **slk_init** is **2** or **3**);

label   is be the string to put on the label, up to eight (five if *fmt* in **slk_init** is **2** or **3**) characters in length. A empty string or a null pointer sets up a blank label.

align   is **0**, **1**, or **2**, aligning *label* to the left, center, or right, respectively, within the 8 (5) character cells housing it.

slk_label obtains the string assigned to label number *labnum*, with any leading and trailing blanks stripped.

Screen Updates

slk_refresh and **slk_noutrefresh** affect the soft key label lines as **wrefresh** and **wnoutrefresh** do the *curses* window.

The **slk_clear** routine clears the soft labels from the screen.

The **slk_restore** routine restores the soft labels to the screen after a **slk_clear** has been performed.

The **slk_touch** routine forces all the soft labels to be output the next time a **slk_noutrefresh** is performed.

Video Attributes

The **slk_attron**, **slk_attrset**, **slk_attroff**, and **slk_attr** routines correspond to **attron**, **attrset**, **attroff**, and **attr_get**, respectively. They have an effect only if soft labels are simulated on the bottom line of the screen. The default highlight for soft key labels is **A_STANDOUT** (as in System V *curses*, which does not document this fact).

Colors

The **slk_color** routine corresponds to **color_set**. It has an effect only if soft labels are simulated on the bottom line of the screen.

Because **slk_color** accepts only *short* (signed 16-bit integer) values, this implementation provides **extended_slk_color**, which accepts an *int* value of at least 32 bits.

RETURN VALUE

Routines that return an integer return **ERR** upon failure and **OK** (SVr4 specifies only "an integer value other than **ERR**") upon successful completion.

X/Open Curses defines no error conditions.

In this implementation

slk_attr
> returns the attribute used for the soft keys.

slk_attroff, **slk_attron**, **slk_clear**, **slk_noutrefresh**, **slk_refresh**, **slk_touch**
> return an error if the terminal or the softkeys were not initialized.

slk_attrset
> returns an error if the terminal or the softkeys were not initialized.

slk_attr_set
> returns an error if the terminal or the softkeys were not initialized, or the color pair is outside the range 0..**COLOR_PAIRS**-1.

slk_color
> returns an error if the terminal or the softkeys were not initialized, or the color pair is outside the range 0..**COLOR_PAIRS**-1.

slk_init

   returns an error if the format parameter is outside the range 0..3.

slk_label

   returns **NULL** on error.

slk_set

   returns an error if the terminal or the softkeys were not initialized, or the *labnum* parameter is
   outside the range of label counts, or if the format parameter is outside the range 0..2, or if
   memory for the labels cannot be allocated.

NOTES

Most applications would use **slk_noutrefresh** because a **wrefresh** is likely to follow soon.

EXTENSIONS

X/Open Curses documents the *opts* argument as reserved for future use, saying that it must be null.
This implementation uses that parameter in ABI 6 for the functions which have a color pair parameter
to support extended color pairs.

For functions which modify the color, e.g., **slk_attr_set**, if *opts* is set it is treated as a pointer to *int*, and
used to set the color pair instead of the *short* pair parameter.

PORTABILITY

X/Open Curses, Issue 4 describes these functions, with some differences from SVr4 *curses*:

⊕   X/Open added functions like the SVr4 attribute-manipulation functions **slk_attron**, **slk_attroff**, and
    slk_attrset, but which use *attr_t* parameters (rather than *chtype*), along with a reserved *opts*
    parameter.

    Two of these new functions (unlike the SVr4 functions) have no provision for color: **slk_attr_on**
    and **slk_attr_off**.

    The third function (**slk_attr_set**) has a color pair parameter.

⊕   It added *const* qualifiers to parameters (unnecessarily), and

⊕   It added **slk_color**.

Although **slk_start** is declared in the *curses* header file, it was not documented by SVr4 other than its
presence in a list of libtermlib.so.1 symbols. Reading the source code (i.e., Illumos):

⊕   **slk_start** has two parameters:

  ⊕   *ng* (number of groups) and

  ⊕   *gp* (group pointer).

⊕   Soft-key groups are an array of *ng* integers.

⊕   In SVr4, **slk_init** calls **slk_start** passing a null for *gp*. For this case, **slk_start** uses the number of groups *ng* (3 for the 3-2-3 layout, 2 for the 4-4 layout) which **slk_init** provided.

   If *ng* is neither 2 or 3, **slk_start** checks the terminfo **fln** (label_format) capability, interpreting that as a comma-separated list of numbers, e.g., "3,2,3" for the 3-2-3 layout.

   Finally, if there is no **fln** capability, **slk_start** returns **ERR**.

⊕   If **slk_start** is given a non-null *gp*, it copies the *ng* elements of the group of soft-keys, up to 16.

   If there are more than 16 elements, **slk_start** returns an error.

⊕   The format codes **2** and **3** for **slk_init** were added by *ncurses* in 1996. PDCurses 2.4 added this feature in 2001.

The function **slk_attr** was added by *ncurses* in 1996.

X/Open Curses does not specify a limit for the number of colors and color pairs which a terminal can support. However, in its use of *short* for the parameters, it carries over SVr4's implementation detail for the compiled terminfo database, which uses signed 16-bit numbers. This implementation provides extended versions of those functions which use *int* parameters, allowing applications to use larger color- and pair-numbers.

HISTORY
SVr3 introduced these functions:
  slk_clear
  slk_init
  slk_label
  slk_noutrefresh
  slk_refresh
  slk_restore
  slk_set

slk_touch

SVr4 added these functions:
slk_attroff
slk_attron
slk_attrset
slk_start

X/Open Curses added these:
slk_attr_off
slk_attr_on
slk_attr_set
slk_color
slk_wset

SEE ALSO

curses(3X), **curs_attr**(3X), **curs_initscr**(3X), **curs_refresh**(3X), **curs_variables**(3X)