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 routines

SYNOPSIS

```
#include <curses.h>

int slk_init(int fmt);

int slk_set(int labnum, const char *label, int fmt);
/* extension */
int slk_wset(int labnum, const wchar_t *label, int fmt);

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);

attr_t slk_attr(void);

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

DESCRIPTION

The slk* functions manipulate the set of soft function-key labels that exist on many terminals. For those terminals that do not have soft labels, **curses** takes over the bottom line of **stdscr**, reducing the size of **stdscr** and the variable **LINES**. **curses** standardizes on eight labels of up to eight characters each. In addition to this, the ncurses implementation supports a mode where it simulates 12 labels of

up to five characters each. This is useful for PC-like enduser devices. 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

The **slk_init** routine must be called before **initscr** or **newterm** is called. 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 identify the key numbers easily.

Labels

The **slk_set** routine (and the **slk_wset** routine for the wide-character library) has three parameters:

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

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

fmt is either **0**, **1**, or **2**, indicating whether the label is to be left-justified, centered, or right-justified, respectively, within the label.

The **slk_label** routine returns the current label for label number *labnum*, with leading and trailing blanks stripped.

Screen updates

The **slk_refresh** and **slk_noutrefresh** routines correspond to the **wrefresh** and **wnoutrefresh** routines.

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 keys 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 integer value, e.g., 32-bits.

RETURN VALUE

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

X/Open 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.

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

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.

NOTES

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

PORTABILITY

The XSI Curses standard, Issue 4, described the soft-key functions, with some differences from SVr4 curses:

- It added functions like the SVr4 attribute-manipulation functions **slk_attron**, **slk_attroff**, **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**.

The format codes **2** and **3** for **slk_init** and the function **slk_attr** are specific to ncurses.

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 **short** parameters, allowing applications to use larger color- and pair-numbers.

SEE ALSO

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