

**NAME**

**slk\_init**, **slk\_set**, **slk\_wset**, **slk\_refresh**, **slk\_noutrefresh**, **slk\_label**, **slk\_clear**, **slk\_restore**, **slk\_touch**, **slk\_atron**, **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_atron(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)**