NAME

bool, *chtype*, *cchar_t*, *attr_t*, *SCREEN*, *WINDOW*, **TRUE**, **FALSE**, **ERR**, **OK**, **curscr**, **newscr**, **stdscr**, **COLORS**, **COLOR_PAIRS**, **COLS**, **LINES**, **ESCDELAY**, **TABSIZE** - *curses* data types, constants, and global variables

SYNOPSIS

```
#include <curses.h>

/* data types */
typedef /* ... */ bool;
typedef /* ... */ chtype;
typedef /* ... */ cchar_t;
typedef /* ... */ attr_t;
typedef /* ... */ SCREEN;
typedef /* ... */ WINDOW;

/* constants */
const bool TRUE;
const bool FALSE;

const /* ... */ ERR;
const /* ... */ OK;

/* variables */
int COLORS;
int COLOR_PAIRS;
int COLS;
int LINES;
WINDOW * curscr;
WINDOW * stdscr;

/* extensions */
int ESCDELAY;
int TABSIZE;
WINDOW * newscr;
```

DESCRIPTION

This page summarizes data types, constants, and variables provided by the *curses* library. Locate further discussion in **curses**(3X).

Depending on *ncurses*'s build-time configuration, the variables may instead be macros (see **curs_threads**(3X) and **curs_opaque**(3X)) that provide read-only access to the library's state. In either case, applications should treat them as read-only to avoid confusing the library.

CONSTANTS

TRUE, FALSE

The *curses* library defines **TRUE** and **FALSE** to represent the values of the Boolean data type.

ERR, OK

curses and *terminfo* routines frequently return these constant integral values indicating failure and success, respectively.

PREDEFINED TYPES

bool

X/Open Issue 4 *curses* (1996) preceded the ISO C99 and ISO C++98 standards, each of which also defined a Boolean data type. The *curses* library requires an integral type *bool*.

ncurses' configure script attempts to discover the data type used by the system's C and C++ compilers, to reuse for the *curses bool*.

chtype

The *chtype* integral type combines a ("narrow", 8-bit) character with attributes encoding the character's *rendition*, such as the styling of its typeface and/or foreground and background colors. See, for example, **addch**(3X), **attron**(3X), and **inch**(3X).

*cchar_t***,** *attr_t*

chtype is too small for the standard C library's wide-character type, *wchar_t*. *cchar_t* is a type that can accommodate an *attr_t* and enough wide characters to store what Unicode terms a *grapheme cluster* (a "user-perceived character" [UAX #29], which may nevertheless require several character encoding units to represent). *attr_t* is an integral type storing "wide" attributes that apply to *cchar_t*s. See, for example, **add_wch**(3X), **attr_on**(3X), and **in_wch**(3X).

SCREEN

curses manages a terminal device with this structure type; see **initscr**(3X).

WINDOW

curses represents rectangular portions of the terminal screen with the *WINDOW* structure type; see subsection "Overview" of **ncurses**(3X).

VARIABLES

curscr, stdscr, newscr

The library records updates to the terminal screen in a window named **curscr**. This object is referred to as the "physical screen" in **curs_refresh**(3X) and **curs_outopts**(3X).

ncurses collects pending updates to the terminal screen in a window named **newscr**. This object is referred to as the "virtual screen" in the **curs_kernel**(3X), **curs_refresh**(3X), and **curs_outopts**(3X). When the screen is refreshed, *curses* determines a minimal set of updates using the terminal's capabilities to make **curscr** look like **newscr**.

Once *curses* is initialized, it creates a window named **stdscr**. It is the same size as the terminal screen and is the default window used by routines that do not take a parameter identifying one. Many *curses* functions use this window.

COLORS

Once *curses* is initialized, **COLORS** contains the number of colors supported by the terminal; see **curs_color**(3X).

COLOR_PAIRS

Once *curses* is initialized, **COLOR_PAIRS** contains the number of color pairs supported by the terminal; see **curs_color**(3X).

COLS, LINES

Once *curses* is initialized, **COLS** and **LINES** contain the screen's width and height in character cells, respectively; that is, the number of columns and lines.

ESCDELAY

For *curses* to distinguish the ESC character resulting from a user's press of the "Escape" key on the input device from one beginning an *escape sequence* (as commonly produced by function keys), it waits after the escape character to see if further characters are available on the input stream within a short interval. **ESCDELAY** stores this interval in milliseconds.

If **keypad**(3X) is disabled for the *curses* window receiving input, a program must disambiguate escape sequences itself.

TABSIZE

The *curses* library converts a tab character to this number of spaces as it adds a tab to a window; see **curs_addch**(3X).

NOTES

Either **initscr**(3X) or **newterm**(3X) initializes *curses*.

If *ncurses* is configured to provide separate *curses* and *tinfo* libraries, most of these variables reside in the former.

PORTABILITY

The X/Open Curses standard documents all of the foregoing types and symbols except for **newscr**, **TABSIZE**, and **ESCDELAY**.

X/Open Curses describes **curscr** only as "an internal data structure"; SVr4 gave more details, noting its use "for certain low-level operations like clearing and redrawing a screen containing garbage". Neither specified its interaction with the rest of the interface beyond use as an argument to **clearok**(3X) and **wrefresh**(3X).

newscr is a feature of SVr4 *curses*. When refreshing the screen, it is used as a working area for combining the standard window **stdscr** with any others the application may have created with **newwin**(3X). When the update of **newscr** is complete, *curses* modifies **curscr** to match **newscr**.

TABSIZE is a feature of SVr4 *curses*.

- SVr4 initially sets **TABSIZE** from the terminal description’s **init_tabs** capability. After that, it can be altered by applications using SVr4 *curses*.

- SVr4 *curses* uses the value of **TABSIZE** to compute the position of tab stops when updating both the virtual screen with **addch**(3X) and the physical screen with **mvcur**(3X).

- *ncurses* uses the value of **TABSIZE** only to update the virtual screen. It uses the terminal description’s "**it**" (**init_tabs**) capability for computing hardware tabs (that is, tab stops on the physical screen).

- Other implementations differ. For instance, NetBSD *curses* allows **TABSIZE** to be set through an environment variable. *ncurses* does not.

  NetBSD *curses* does not support hardware tabs; it uses the **init_tabs** capability and the **TABSIZE** variable only for updating the virtual screen.

ESCDELAY is a feature of AIX *curses*.

- In AIX, the units for **ESCDELAY** are *fifths* of milliseconds.

- The default value for AIX’s **ESCDELAY** equals 0.1 seconds.

- AIX also enforces a limit of 10,000 seconds for **ESCDELAY**; *ncurses* does not enforce any upper limit.

ncurses has long used **ESCDELAY** with units of milliseconds, making it impossible to be completely compatible with AIX. Consequently, most users have decided either to override the value, or to rely upon its default.

SEE ALSO

curses(3X), **curs_color**(3X), **curs_opaque**(3X), **curs_terminfo**(3X), **curs_threads**(3X), **term_variables**(3X), **terminfo**(5)

[UAX #29] "Unicode Standard Annex #29: Unicode Text Segmentation"; <https://unicode.org/reports/tr29/>