

NAME

SP, **acs_map**, **boolcodes**, **boolfnames**, **boolnames**, **cur_term**, **numcodes**, **numfnames**, **numnames**, **strcodes**, **strfnames**, **strnames**, **ttytype** – *terminfo* global variables

SYNOPSIS

```
#include <curses.h>
#include <term.h>

chtype acs_map[];
SCREEN * SP;
TERMINAL * cur_term;
char ttytype[];

NCURSES_CONST char * const boolcodes[];
NCURSES_CONST char * const boolfnames[];
NCURSES_CONST char * const boolnames[];

NCURSES_CONST char * const numcodes[];
NCURSES_CONST char * const numfnames[];
NCURSES_CONST char * const numnames[];

NCURSES_CONST char * const strcodes[];
NCURSES_CONST char * const strfnames[];
NCURSES_CONST char * const strnames[];
```

DESCRIPTION

This page summarizes variables provided by the *curses* library's lower-level *terminfo* interface. Locate a more complete description in the **cur_terminfo(3X)** manual page.

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

Alternate Character Set Mapping

After initializing the *curses* or *terminfo* interfaces, the **acs_map** array holds information used to translate cells with the **A_ALTCHARSET** video attribute into line-drawing characters.

The encoding of the information in this array has changed periodically. Application developers need only know that it is used for the "ACS_" constants in *ncurses.h*.

The comparable data for the wide-character library are not exposed as symbols in the API.

Current Terminal Data

After initializing the *curses* or *terminfo* interfaces, **cur_term** contains data describing the current terminal. It is also updated as a side effect of **set_term(3X)** and **delscreen(3X)**.

It is possible to save a value of **cur_term** for subsequent use as a parameter to **set_term(3X)** for switching between screens. Alternatively, one can save the return value from **newterm(3X)** or **setupterm(3X)** to reuse in **set_term(3X)**.

terminfo Lookup Tables

The **tic(1)** and **infocmp(1)** programs use lookup tables for the long and short names of *terminfo* capabilities, as well as the corresponding names for *termcap* capabilities. These are available to other applications, though the hash tables used by the *terminfo* and *termcap* functions are not.

terminfo stores capability names in arrays with an "f" (eff) in their names: **boolfnames**, **numfnames**, and **strfnames**. It stores the briefer capability codes in arrays without the "f": **boolnames**, **numnames**, and **strnames**. The corresponding *termcap* capability codes are stored in **boolcodes**, **numcodes**, and **strcodes**. **terminfo(5)** catalogs these.

Terminal Type

A terminal description begins with one or more terminal names separated by “|” (vertical bars). On initialization of the *curses* or *terminfo* interfaces, **setupterm(3X)** copies the terminal name to the array **ttytype**.

terminfo Names

In addition to the variables, *term.h* also defines a symbol for each *terminfo* capability name. These are in terms of the symbol **CUR**, which is defined

```
#define CUR ((TERMTYPE *)(cur_term))->
```

These symbols provide a faster method of accessing *terminfo* capabilities than using, for example, **tigetstr(3X)**.

The definition of **CUR** is implementation-dependent, but each *terminfo* library defines these names to point into the in-memory description of the current terminal. For example, in *ncurses*, the expressions **tigetstr("key_dc")** and “**CUR Strings[59]**” are equivalent.

terminfo and curses

setupterm(3X) initializes the lower-level *terminfo* interface. The higher-level *curses* interface uses the *terminfo* interface internally; the symbol **SP** bridges the *curses* *SCREEN* type with the *terminfo* *TERMINAL* type.

PORTABILITY

X/Open Curses does not describe any of these symbols except for *cur_term*. (The inclusion of *cur_term* appears to be an oversight, since other comparable low-level information is omitted by X/Open.)

Other implementations may have comparable variables. Some implementations provide the variables in their libraries, but omit them from the header files.

All implementations that provide *terminfo* interfaces add definitions as described in the “*terminfo* Names” section above. Most, but not all, base the definition upon the *cur_term* variable.

SEE ALSO

curses(3X), **curs_terminfo(3X)**, **curs_threads(3X)**, **terminfo(5)**