

NAME

SP, **acs_map**, **boolcodes**, **boolfnames**, **boolnames**, **cur_term**, **numcodes**, **numfnames**, **numnames**, **strcodes**, **strfnames**, **strnames**, **ttytype** - **curses** 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 low-level terminfo interface. A more complete description is given in the **curl_terminfo(3X)** manual page.

Depending on the configuration, these may be actual variables, or macros (see **curl_threads(3X)**) which 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 <curses.h>.

The comparable data for the wide-character library is a private variable.

Current Terminal Data

After initializing the curses or terminfo interfaces, the **cur_term** contains data describing the current terminal. This variable is also set 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**, for switching between screens. Alternatively, one can save the return value from **newterm** or **setupterm(3X)** to reuse in **set_term**.

Terminfo Names

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, although the hash-tables used by the terminfo and termcap functions are not available.

The long terminfo capability names use a "l" (ell) in their names: **boolfnames**, **numfnames**, and **strfnames**.

These are the short names for terminfo capabilities: **boolnames**, **numnames**, and **strnames**.

These are the corresponding names used for termcap descriptions: **boolcodes**, **numcodes**, and **strcodes**.

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 names to the array **ttytype**.

Terminfo Names

In addition to the variables, <**term.h**> also defines a symbol for each terminfo capability *long 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 **tigetstr(3X)**, etc.

The actual definition of **CUR** depends upon the implementation, but each terminfo library provides these long names defined to point into the current terminal description loaded into memory.

NOTES

The low-level terminfo interface is initialized using **setupterm**(3X). The upper-level curses interface uses the low-level terminfo interface, internally.

PORTABILITY

X/Open Curses does not describe any of these 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 which provide terminfo interfaces add definitions as described in the **Terminfo Names** section. Most, but not all, base the definition upon the **cur_term** variable.

SEE ALSO

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