NAME

PC, **UP**, **BC**, **ospeed**, **tgetent**, **tgetflag**, **tgetnum**, **tgetstr**, **tgoto**, **tputs - curses** emulation of termcap

SYNOPSIS

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

extern char PC;
extern char * UP;
extern char * BC;
extern short ospeed;

int tgetent(char *bp, const char *name);
int tgetflag(const char *id);
int tgetnum(const char *id);
char *tgetstr(const char *id, char **area);
char *tgoto(const char *cap, int col, int row);
int tputs(const char *str, int affcnt, int (*putc)(int));
```

DESCRIPTION

These routines are included as a conversion aid for programs that use the *termcap* library. Their parameters are the same, but the routines are emulated using the *terminfo* database. Thus, they can only be used to query the capabilities of entries for which a terminfo entry has been compiled.

INITIALIZATION

The **tgetent** routine loads the entry for *name*. It returns:

1 on success,

0 if there is no such entry (or that it is a generic type, having too little information for curses applications to run), and

-1 if the terminfo database could not be found.

This differs from the *termcap* library in two ways:

- The emulation ignores the buffer pointer *bp*. The *termcap* library would store a copy of the terminal description in the area referenced by this pointer. However, ncurses stores its terminal descriptions in compiled binary form, which is not the same thing.

- There is a difference in return codes. The *termcap* library does not check if the terminal description is marked with the *generic* capability, or if the terminal description has cursor-addressing.

CAPABILITY VALUES

The **tgetflag** routine gets the boolean entry for *id*, or zero if it is not available.

The **tgetnum** routine gets the numeric entry for *id*, or -1 if it is not available.

The **tgetstr** routine returns the string entry for *id*, or zero if it is not available. Use **tputs** to output the returned string. The *area* parameter is used as follows:

- It is assumed to be the address of a pointer to a buffer managed by the calling application.
- However, ncurses checks to ensure that **area** is not NULL, and also that the resulting buffer pointer is not NULL. If either check fails, the *area* parameter is ignored.
- If the checks succeed, ncurses also copies the return value to the buffer pointed to by *area*, and the *area* value will be updated to point past the null ending this value.
- The return value itself is an address in the terminal description which is loaded into memory.

Only the first two characters of the **id** parameter of **tgetflag**, **tgetnum** and **tgetstr** are compared in lookups.

FORMATTING CAPABILITIES

The **tgoto** routine expands the given capability using the parameters.

- Because the capability may have padding characters, the output of **tgoto** should be passed to **tputs** rather than some other output function such as **printf**.
- While **tgoto** is assumed to be used for the two-parameter cursor positioning capability, termcap applications also use it for single-parameter capabilities.

  Doing this shows a quirk in **tgoto**: most hardware terminals use cursor addressing with *row* first, but the original developers of the termcap interface chose to put the *column* parameter first. The **tgoto** function swaps the order of parameters. It does this also for calls requiring only a single parameter. In that case, the first parameter is merely a placeholder.
- Normally the ncurses library is compiled with terminfo support. In that case, **tgoto** uses

tparm(3X) (a more capable formatter).

However, **tparm** is not a *termcap* feature, and portable *termcap* applications should not rely upon its availability.

The **tputs** routine is described on the **curs_terminfo**(3X) manual page. It can retrieve capabilities by either termcap or terminfo name.

GLOBAL VARIABLES

The variables **PC**, **UP** and **BC** are set by **tgetent** to the terminfo entry's data for **pad_char**, **cursor_up** and **backspace_if_not_bs**, respectively. **UP** is not used by ncurses. **PC** is used in the **tdelay_output** function. **BC** is used in the **tgoto** emulation. The variable **ospeed** is set by ncurses in a system-specific coding to reflect the terminal speed.

RETURN VALUE

Except where explicitly noted, routines that return an integer return **ERR** upon failure and **OK** (SVr4 only specifies "an integer value other than **ERR**") upon successful completion.

Routines that return pointers return **NULL** on error.

BUGS

If you call **tgetstr** to fetch **ca** or any other parameterized string, be aware that it will be returned in terminfo notation, not the older and not-quite-compatible termcap notation. This will not cause problems if all you do with it is call **tgoto** or **tparm**, which both expand terminfo-style strings as terminfo. (The **tgoto** function, if configured to support termcap, will check if the string is indeed terminfo-style by looking for "%p" parameters or "$<..>" delays, and invoke a termcap-style parser if the string does not appear to be terminfo).

Because terminfo conventions for representing padding in string capabilities differ from termcap's, **tputs("50");** will put out a literal "50" rather than busy-waiting for 50 milliseconds. Cope with it.

Note that termcap has nothing analogous to terminfo's **sgr** string. One consequence of this is that termcap applications assume me (terminfo **sgr0**) does not reset the alternate character set. This implementation checks for, and modifies the data shown to the termcap interface to accommodate termcap's limitation in this respect.

PORTABILITY

Standards

These functions are provided for supporting legacy applications, and should not be used in new programs:

- The XSI Curses standard, Issue 4 describes these functions. However, they are marked TO BE WITHDRAWN and may be removed in future versions.

- X/Open Curses, Issue 5 (December 2007) marked the termcap interface (along with **vwprintw** and **vwscanw**) as withdrawn.

Neither the XSI Curses standard nor the SVr4 man pages documented the return values of **tgetent** correctly, though all three were in fact returned ever since SVr1. In particular, an omission in the XSI Curses documentation has been misinterpreted to mean that **tgetent** returns **OK** or **ERR**. Because the purpose of these functions is to provide compatibility with the *termcap* library, that is a defect in XCurses, Issue 4, Version 2 rather than in ncurses.

Compatibility with BSD Termcap

External variables are provided for support of certain termcap applications. However, termcap applications' use of those variables is poorly documented, e.g., not distinguishing between input and output. In particular, some applications are reported to declare and/or modify **ospeed**.

The comment that only the first two characters of the **id** parameter are used escapes many application developers. The original BSD 4.2 termcap library (and historical relics thereof) did not require a trailing null NUL on the parameter name passed to **tgetstr**, **tgetnum** and **tgetflag**. Some applications assume that the termcap interface does not require the trailing NUL for the parameter name. Taking into account these issues:

- As a special case, **tgetflag** matched against a single-character identifier provided that was at the end of the terminal description. You should not rely upon this behavior in portable programs. This implementation disallows matches against single-character capability names.

- This implementation disallows matches by the termcap interface against extended capability names which are longer than two characters.

The BSD termcap function **tgetent** returns the text of a termcap entry in the buffer passed as an argument. This library (like other terminfo implementations) does not store terminal descriptions as text. It sets the buffer contents to a null-terminated string.

Other Compatibility

This library includes a termcap.h header, for compatibility with other implementations. But the header is rarely used because the other implementations are not strictly compatible.

The original BSD termcap (through 4.3BSD) had no header file which gave function prototypes, because that was a feature of ANSI C. BSD termcap was written several years before C was

standardized. However, there were two different termcap.h header files in the BSD sources:

- One was used internally by the *jove* editor in 2BSD through 4.4BSD. It defined global symbols for the termcap variables which it used.

- The other appeared in 4.4BSD Lite Release 2 (mid-1993) as part of *libedit* (also known as the *editline* library). The CSRG source history shows that this was added in mid-1992. The *libedit* header file was used internally, as a convenience for compiling the *editline* library. It declared function prototypes, but no global variables.

The header file from *libedit* was added to NetBSD's termcap library in mid-1994.

Meanwhile, GNU termcap was under development, starting in 1990. The first release (termcap 1.0) in 1991 included a termcap.h header. The second release (termcap 1.1) in September 1992 modified the header to use **const** for the function prototypes in the header where one would expect the parameters to be read-only. This was a difference versus the original BSD termcap. The prototype for **tputs** also differed, but in that instance, it was *libedit* which differed from BSD termcap.

A copy of GNU termcap 1.3 was bundled with *bash* in mid-1993, to support the *readline* library.

A termcap.h file was provided in ncurses 1.8.1 (November 1993). That reflected influence by *emacs* (rather than *jove*) and GNU termcap:

- it provided declarations for a few global symbols used by *emacs*

- it provided function prototypes (using **const**).

- a prototype for **tparam** (a GNU termcap feature) was provided.

Later (in mid-1996) the **tparam** function was removed from ncurses. As a result, there are differences between any of the four implementations, which must be taken into account by programs which can work with all termcap library interfaces.

SEE ALSO

curses(3X), **putc**(3), **term_variables**(3X), **terminfo**(5).

https://invisible-island.net/ncurses/tctest.html