NAME

tctest – termcap library checker

SYNOPSIS

tctest *[options] [file-specifications]*

DESCRIPTION

tctest exercises the termcap library (or emulation of termcap) with which it is linked. It provides several command-line options, making it simple to construct test-cases to compare implementations of termcap.

OPTIONS

After processing options, **tctest** uses any command-line parameters as the names of terminal descriptions to process. If no command-line parameters are given, **tctest** tries the **TERM** environment variable. If that is not set, the program uses the **dumb** terminal description.

The termcap library function **tgetent** locates the termcap file, looking at the **TERMCAP** environment variable (or **/etc/termcap**). **tctest** manipulates this environment variable, allowing one to use alternate terminal databases.

Some of the options provide only summary behavior. The most useful output from **tctest** is in *termcap* format, with one capability per line. It is noted that the extra whitespace would be poorly handled by some termcap implementations, counting toward the 1023 byte limit on an entry's size. The reason for listing it in this way is to provide easy comparison.

These are the options:

–1 when processing a termcap file, e.g., with **–a**, call **tgetent** only once per entry using the primary name. Use this for getting per-entry statistics versus the per-call statistics.

–a show capabilities for all names in termcap file. This overrides the **–b** option.

–b use brute-force to find all capabilities, e.g., for the names given on the command-line.

–c **tctest** uses the names from a built-in table of termcap names chosen to correspond to terminfo capability names. This is the default; the **–c** option is provided to cancel **–b**.

–e use $TERMCAP variable if it exists. If this option is not set, **tctest** unsets the **TERMCAP** environment variable.

–f *NAME*
use this termcap file. **tctest** sets the **TERMCAP** environment variable to the file's absolute path.

–g writes two files (**by-name.dat** and **by-size.dat**) with statistics gathered while processing other options, e.g., **–a** or **–s**):

by-name.dat columns

1 the 2-character capability name in double-quotes

2 total number of times the capability was retrieved

3 number denoting if the value is "standard" (1) or "obsolete" (3).

by-size.dat columns

1 size of the buffer returned by **tgetent**

2 total number of **tgetent** calls returning this size

3 total wasted space (tabs, newlines) for **tgetent** calls returning this size

–l list names and aliases in termcap file.

–n do not query for the capability names. This option is used to allow timing "only" **tgetent**.

–o *NAME*
redirect output (the reformatted input) to this file. By default that goes to the standard output. If this option is used, verbose and summary reports are written to the standard output instead of the standard error, to allow separating them from error messages that may be written by the termcap

library.

−r *COUNT*
repeat the processing of the terminal database this many times. As in **−n**, this is used to help measure timing for **tgetent**.

−s show a summary report.

−v verbose (prints names to stderr to track tgetent calls). This is useful in conjunction with **−a**, e.g., to help pinpoint places where **tgetent** reports that a terminal entry is too large. Repeat the option to get more detail.

−w check the result from **tgetent** and **tgetstr**, printing warnings to the standard error.

−V print the program version and exit

SEE ALSO

tgetent(3) documentation for ncurses highlights the differences. This is a starting point; although the original BSD source code is recommended reading as well.

ncurses(3) is a terminfo-based library, which provides a termcap interface. Normally it ignores the **TERMCAP** variable, but it can be configured to provide a more accurate emulation of the termcap library, e.g., loading the terminal entries base on **TERMCAP**.

AUTHOR

Thomas Dickey <dickey@invisible-island.net>.