NAME

termcap − terminal capability database

DESCRIPTION

The termcap database is an obsolete facility for describing the capabilities of character-cell terminals and printers. It is retained only for compatibility with old programs; new programs should use the **terminfo**(5) database and associated libraries.

/etc/termcap is an ASCII file (the database master) that lists the capabilities of many different types of terminals. Programs can read termcap to find the particular escape codes needed to control the visual attributes of the terminal actually in use. (Other aspects of the terminal are handled by **stty**(1).) The termcap database is indexed on the **TERM** environment variable.

Termcap entries must be defined on a single logical line, with '\' used to suppress the newline. Fields are separated by ':'. The first field of each entry starts at the left-hand margin, and contains a list of names for the terminal, separated by '|'.

The first subfield may (in BSD termcap entries from 4.3BSD and earlier) contain a short name consisting of two characters. This short name may consist of capital or small letters. In 4.4BSD, termcap entries this field is omitted.

The second subfield (first, in the newer 4.4BSD format) contains the name used by the environment variable **TERM**. It should be spelled in lowercase letters. Selectable hardware capabilities should be marked by appending a hyphen and a suffix to this name. See below for an example. Usual suffixes are w (more than 80 characters wide), am (automatic margins), nam (no automatic margins), and rv (reverse video display). The third subfield contains a long and descriptive name for this termcap entry.

Subsequent fields contain the terminal capabilities; any continued capability lines must be indented one tab from the left margin.

Although there is no defined order, it is suggested to write first boolean, then numeric, and then string capabilities, each sorted alphabetically without looking at lower or upper spelling. Capabilities of similar functions can be written in one line.

Example for:

```
Head line: vt|vt101|DEC VT 101 terminal in 80 character mode:\
Head line: Vt|vt101-w|DEC VT 101 terminal in (wide) 132 character mode:\
Boolean: :bs:\
Numeric: :co#80:\
String: :sr=\E[H:\
```

Boolean capabilities

5i	Printer will not echo on screen
am	Automatic margins which means automatic line wrap
bs	Control-H (8 dec.) performs a backspace
bw	Backspace on left margin wraps to previous line and right margin
da	Display retained above screen
db	Display retained below screen
eo	A space erases all characters at cursor position
es	Escape sequences and special characters work in status line
gn	Generic device
hc	This is a hardcopy terminal
HC	The cursor is hard to see when not on bottom line
hs	Has a status line
hz	Hazeltine bug, the terminal can not print tilde characters
in	Terminal inserts null bytes, not spaces, to fill whitespace
km	Terminal has a meta key
mi	Cursor movement works in insert mode
ms	Cursor movement works in standout/underline mode

NP	No pad character
NR	ti does not reverse te
nx	No padding, must use XON/XOFF
os	Terminal can overstrike
ul	Terminal underlines although it can not overstrike
xb	Beehive glitch, f1 sends ESCAPE, f2 sends **^C**
xn	Newline/wraparound glitch
xo	Terminal uses xon/xoff protocol
xs	Text typed over standout text will be displayed in standout
xt	Teleray glitch, destructive tabs and odd standout mode

Numeric capabilities

co	Number of columns
dB	Delay in milliseconds for backspace on hardcopy terminals
dC	Delay in milliseconds for carriage return on hardcopy terminals
dF	Delay in milliseconds for form feed on hardcopy terminals
dN	Delay in milliseconds for new line on hardcopy terminals
dT	Delay in milliseconds for tabulator stop on hardcopy terminals
dV	Delay in milliseconds for vertical tabulator stop on hardcopy terminals
it	Difference between tab positions
lh	Height of soft labels
lm	Lines of memory
lw	Width of soft labels
li	Number of lines
Nl	Number of soft labels
pb	Lowest baud rate which needs padding
sg	Standout glitch
ug	Underline glitch
vt	virtual terminal number
ws	Width of status line if different from screen width

String capabilities

!1	shifted save key
!2	shifted suspend key
!3	shifted undo key
#1	shifted help key
#2	shifted home key
#3	shifted input key
#4	shifted cursor left key
%0	redo key
%1	help key
%2	mark key
%3	message key
%4	move key
%5	next-object key
%6	open key
%7	options key
%8	previous-object key
%9	print key
%a	shifted message key
%b	shifted move key
%c	shifted next key
%d	shifted options key
%e	shifted previous key
%f	shifted print key

%g	shifted redo key
%h	shifted replace key
%i	shifted cursor right key
%j	shifted resume key
&0	shifted cancel key
&1	reference key
&2	refresh key
&3	replace key
&4	restart key
&5	resume key
&6	save key
&7	suspend key
&8	undo key
&9	shifted begin key
*0	shifted find key
*1	shifted command key
*2	shifted copy key
*3	shifted create key
*4	shifted delete character
*5	shifted delete line
*6	select key
*7	shifted end key
*8	shifted clear line key
*9	shifted exit key
@0	find key
@1	begin key
@2	cancel key
@3	close key
@4	command key
@5	copy key
@6	create key
@7	end key
@8	enter/send key
@9	exit key
al	Insert one line
AL	Insert %1 lines
ac	Pairs of block graphic characters to map alternate character set
ae	End alternative character set
as	Start alternative character set for block graphic characters
bc	Backspace, if not **^H**
bl	Audio bell
bt	Move to previous tab stop
cb	Clear from beginning of line to cursor
cc	Dummy command character
cd	Clear to end of screen
ce	Clear to end of line
ch	Move cursor horizontally only to column %1
cl	Clear screen and cursor home
cm	Cursor move to row %1 and column %2 (on screen)
CM	Move cursor to row %1 and column %2 (in memory)
cr	Carriage return
cs	Scroll region from line %1 to %2
ct	Clear tabs
cv	Move cursor vertically only to line %1

dc	Delete one character
DC	Delete %1 characters
dl	Delete one line
DL	Delete %1 lines
dm	Begin delete mode
do	Cursor down one line
DO	Cursor down #1 lines
ds	Disable status line
eA	Enable alternate character set
ec	Erase %1 characters starting at cursor
ed	End delete mode
ei	End insert mode
ff	Formfeed character on hardcopy terminals
fs	Return character to its position before going to status line
F1	The string sent by function key f11
F2	The string sent by function key f12
F3	The string sent by function key f13
...	...
F9	The string sent by function key f19
FA	The string sent by function key f20
FB	The string sent by function key f21
...	...
FZ	The string sent by function key f45
Fa	The string sent by function key f46
Fb	The string sent by function key f47
...	...
Fr	The string sent by function key f63
hd	Move cursor a half line down
ho	Cursor home
hu	Move cursor a half line up
i1	Initialization string 1 at login
i3	Initialization string 3 at login
is	Initialization string 2 at login
ic	Insert one character
IC	Insert %1 characters
if	Initialization file
im	Begin insert mode
ip	Insert pad time and needed special characters after insert
iP	Initialization program
K1	upper left key on keypad
K2	center key on keypad
K3	upper right key on keypad
K4	bottom left key on keypad
K5	bottom right key on keypad
k0	Function key 0
k1	Function key 1
k2	Function key 2
k3	Function key 3
k4	Function key 4
k5	Function key 5
k6	Function key 6
k7	Function key 7
k8	Function key 8
k9	Function key 9

k;	Function key 10
ka	Clear all tabs key
kA	Insert line key
kb	Backspace key
kB	Back tab stop
kC	Clear screen key
kd	Cursor down key
kD	Key for delete character under cursor
ke	turn keypad off
kE	Key for clear to end of line
kF	Key for scrolling forward/down
kh	Cursor home key
kH	Cursor hown down key
kI	Insert character/Insert mode key
kl	Cursor left key
kL	Key for delete line
kM	Key for exit insert mode
kN	Key for next page
kP	Key for previous page
kr	Cursor right key
kR	Key for scrolling backward/up
ks	Turn keypad on
kS	Clear to end of screen key
kt	Clear this tab key
kT	Set tab here key
ku	Cursor up key
l0	Label of zeroth function key, if not f0
l1	Label of first function key, if not f1
l2	Label of first function key, if not f2
...	...
la	Label of tenth function key, if not f10
le	Cursor left one character
ll	Move cursor to lower left corner
LE	Cursor left %1 characters
LF	Turn soft labels off
LO	Turn soft labels on
mb	Start blinking
MC	Clear soft margins
md	Start bold mode
me	End all mode like so, us, mb, md, and mr
mh	Start half bright mode
mk	Dark mode (Characters invisible)
ML	Set left soft margin
mm	Put terminal in meta mode
mo	Put terminal out of meta mode
mp	Turn on protected attribute
mr	Start reverse mode
MR	Set right soft margin
nd	Cursor right one character
nw	Carriage return command
pc	Padding character
pf	Turn printer off
pk	Program key %1 to send string %2 as if typed by user
pl	Program key %1 to execute string %2 in local mode

pn	Program soft label %1 to show string %2
po	Turn the printer on
pO	Turn the printer on for %1 (<256) bytes
ps	Print screen contents on printer
px	Program key %1 to send string %2 to computer
r1	Reset string 1 to set terminal to sane modes
r2	Reset string 2 to set terminal to sane modes
r3	Reset string 3 to set terminal to sane modes
RA	disable automatic margins
rc	Restore saved cursor position
rf	Reset string filename
RF	Request for input from terminal
RI	Cursor right %1 characters
rp	Repeat character %1 for %2 times
rP	Padding after character sent in replace mode
rs	Reset string
RX	Turn off XON/XOFF flow control
sa	Set %1 %2 %3 %4 %5 %6 %7 %8 %9 attributes
SA	enable automatic margins
sc	Save cursor position
se	End standout mode
sf	Normal scroll one line
SF	Normal scroll %1 lines
so	Start standout mode
sr	Reverse scroll
SR	scroll back %1 lines
st	Set tabulator stop in all rows at current column
SX	Turn on XON/XOFF flow control
ta	move to next hardware tab
tc	Read in terminal description from another entry
te	End program that uses cursor motion
ti	Begin program that uses cursor motion
ts	Move cursor to column %1 of status line
uc	Underline character under cursor and move cursor right
ue	End underlining
up	Cursor up one line
UP	Cursor up %1 lines
us	Start underlining
vb	Visible bell
ve	Normal cursor visible
vi	Cursor invisible
vs	Standout cursor
wi	Set window from line %1 to %2 and column %3 to %4
XF	XOFF character if not **^S**

There are several ways of defining the control codes for string capabilities:

Every normal character represents itself, except '^', '\', and '%'.

A **^x** means Control-x. Control-A equals 1 decimal.

\x means a special code. x can be one of the following characters:
E Escape (27)
n Linefeed (10)
r Carriage return (13)
t Tabulation (9)
b Backspace (8)

f Form feed (12)
0 Null character. A \xxx specifies the octal character xxx.

i Increments parameters by one.

r Single parameter capability

+ Add value of next character to this parameter and do binary output

2 Do ASCII output of this parameter with a field with of 2

d Do ASCII output of this parameter with a field with of 3

% Print a '%'

If you use binary output, then you should avoid the null character ('\0') because it terminates the string. You should reset tabulator expansion if a tabulator can be the binary output of a parameter.

Warning:
The above metacharacters for parameters may be wrong: they document Minix termcap which may not be compatible with Linux termcap.

The block graphic characters can be specified by three string capabilities:

as start the alternative charset

ae end the alternative charset

ac pairs of characters. The first character is the name of the block graphic symbol and the second characters is its definition.

The following names are available:

+	right arrow (>)
,	left arrow (<)
.	down arrow (v)
0	full square (#)
I	lantern (#)
-	upper arrow (^)
’	rhombus (+)
a	chess board (:)
f	degree (’)
g	plus-minus (#)
h	square (#)
j	right bottom corner (+)
k	right upper corner (+)
l	left upper corner (+)
m	left bottom corner (+)
n	cross (+)
o	upper horizontal line (-)
q	middle horizontal line (-)
s	bottom horizontal line (_)
t	left tee (+)
u	right tee (+)
v	bottom tee (+)
w	normal tee (+)
x	vertical line (|)
~	paragraph (???)

The values in parentheses are suggested defaults which are used by the *curses* library, if the capabilities are missing.

SEE ALSO

ncurses(3), **termcap**(3), **terminfo**(5)