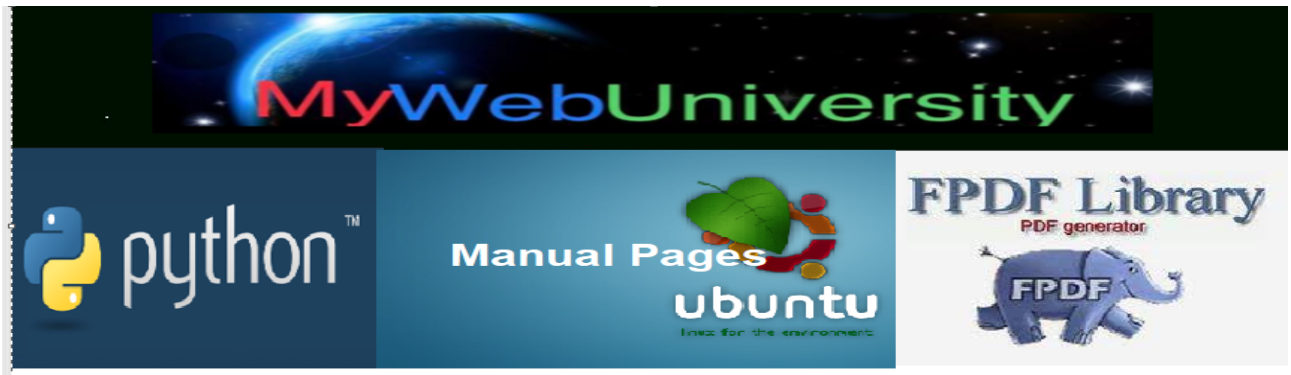

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'infocmp.1'

$ man infocmp.1

infocmp(1) User commands infocmp(1)

NAME

infocmp - compare or print out terminfo descriptions

SYNOPSIS

infocmp [-1cCdDeEFgGIlKlLnpqrtTuUVWx] [-A directory] [-B directory] [-Q encoding] [-R sub?
set] [-s key] [-v level] [-w width] [terminal-type ...]

DESCRIPTION

infocmp reports a human-readable terminal type description from a compiled entry in the terminfo database in a variety of selectable formats, compares such entries to each other, and rewrites an entry to replace ?use? expressions with the content of other entries by ref? erence. A terminfo entry comprises a list of one or more terminal type identifiers, a hu? man-readable description of the terminal type, and a list of terminal capabilities that characterize its programming interface. In all cases, the program reports Boolean-valued capabilities first, followed by numeric ones, and then string-valued capabilities.

Default Options

If no options are specified and zero or one terminal-types is specified, infocmp assumes the -I option. If more than one is specified, the program assumes the -d option.

Source Listing Options [-I] [-L] [-C] [-r]

The -I, -L, and -C options will produce a source listing for each terminal named.

-I use terminfo capability codes

-L use ?long? capability names

-C use termcap capability codes

-r with -C, include nonstandard capabilities

-K with -C, improve BSD compatibility

If no terminal-types are given, the environment variable TERM will be used for the terminal name.

The source produced by the -C option may be used directly as a termcap entry, but not all parameterized strings can be changed to the termcap format. infocmp will attempt to convert most of the parameterized information, and anything not converted will be plainly marked in the output and commented out. These should be edited by hand.

For best results when converting to termcap format, you should use both -C and -r. Normally a termcap description is limited to 1023 bytes. infocmp trims away less essential parts to make it fit. If you are converting to one of the (rare) termcap implementations which ac? cept an unlimited size of termcap, you may want to add the -T option. More often however, you must help the termcap implementation, and trim excess whitespace (use the -0 option for that).

All padding information for strings will be collected together and placed at the beginning of the string where termcap expects it. Mandatory padding (padding information with a trailing ?/?) will become optional.

All termcap variables no longer supported by terminfo, but which are derivable from other terminfo variables, will be output. Not all terminfo capabilities will be translated; only those variables which were part of termcap will normally be output. Specifying the -r op? tion will take off this restriction, allowing all capabilities to be output in termcap form.

Normally you would use both the -C and -r options. The actual format used incorporates some improvements for escaped characters from terminfo format. For a stricter BSD-compatible translation, use the -K option rather than -C.

Note that because padding is collected to the beginning of the capability, not all capabili? ties are output. Mandatory padding is not supported. Because termcap strings are not as flexible, it is not always possible to convert a terminfo string capability into an equiva? lent termcap format. A subsequent conversion of the termcap file back into terminfo format will not necessarily reproduce the original terminfo source.

Some common terminfo parameter sequences, their termcap equivalents, and some terminal types which commonly have such sequences, are:

terminfo	termcap	Terminal Types
??		
%p1%c	%.	ansi-m

%p1%d	%d	ansi, vt100
%p1%' '%+%c	%+x	vt52
%i	%iq	ansi, vt100
%p1%?%'x'%>%t%p1%'y'%+%;	%>xy	annarbor4080
%p2...%p1	%r	hpgeneric

Entry Comparison Options [-d] [-c] [-n]

Given -c, -d, or -n, infocmp compares the terminfo description of the first specified termi? nal-type with those of each of the subsequent operands. If fewer terminal-types than re? quired are specified, infocmp uses the environment variable TERM in their place.

If a capability is defined for only one terminal type, the value reported depends on the ca? pability's type:

? F for missing Boolean variables

? NULL for missing integer or string variables

The -c and -d options report string capability values between ?'? characters. Use the -q option to distinguish absent and canceled capabilities; see terminfo(5).

The comparison option selects the form of report.

-d lists each capability that differs between two entries. Each capability name is fol? lowed by ?:? and comma-separated capability values, then a period.

-c lists each capability that two entries have in common. infocmp ignores capabilities missing from either entry. Each capability name is followed by ?=?, a space, and the capability value, then a period.

If the -u option is further specified, infocmp rewrites the description of the first type employing ?use=? syntax to use the second as a building block.

-n lists capabilities that are in none of the given entries. Each capability name is pre? ceded by ?!? and followed by a period.

Normally only conventional capabilities are shown. Use the -x option to add BSD-com? patibility capabilities (names prefixed with ?OT?).

Use= Option [-u]

The -u option produces a terminfo source description of the first terminal terminal-type which is relative to the sum of the descriptions given by the entries for the other termi? nal-types. It does this by analyzing the differences between the first terminal-types and the other terminal-types and producing a description with use= fields for the other termi? nals. In this manner, it is possible to retrofit generic terminfo entries into a terminal's

description. Or, if two similar terminals exist, but were coded at different times or by different people so that each description is a full description, using infocmp will show what can be done to change one description to be relative to the other.

A capability will be printed with an at-sign (@) if it no longer exists in the first termi? nal-type, but one of the other terminal-type entries contains a value for it. A capabil? ity's value will be printed if the value in the first terminal-type is not found in any of the other terminal-type entries, or if the first of the other terminal-type entries that has this capability gives a different value for the capability than that in the first terminal-type.

The order of the other terminal-type entries is significant. Since the terminfo compiler tic does a left-to-right scan of the capabilities, specifying two use= entries that contain differing entries for the same capabilities will produce different results depending on the order that the entries are given in. infocmp will flag any such inconsistencies between the other terminal-type entries as they are found.

Alternatively, specifying a capability after a use= entry that contains that capability will cause the second specification to be ignored. Using infocmp to recreate a description can be a useful check to make sure that everything was specified correctly in the original source description.

Another error that does not cause incorrect compiled files, but will slow down the compila? tion time, is specifying extra use= fields that are superfluous. infocmp will flag any other terminal-type use= fields that were not needed.

Changing Databases [-A directory] [-B directory]

Like other ncurses utilities, infocmp looks for the terminal descriptions in several places.

You can use the TERMINFO and TERMINFO_DIRS environment variables to override the compiled-in default list of places to search. See ncurses(3NCURSES), as well as the Fetching Compiled Descriptions section in terminfo(5).

You can also use the options -A and -B to override the list of places to search when compar? ing terminal descriptions:

? The -A option sets the location for the first terminal-type

? The -B option sets the location for the other terminal-types.

Using these options, it is possible to compare descriptions for a terminal with the same name located in two different databases. For instance, you can use this feature for compar? ing descriptions for the same terminal created by different people.

Other Options

-0 causes the fields to be printed on one line, without wrapping.

-1 causes the fields to be printed out one to a line. Otherwise, the fields will be printed several to a line to a maximum width of 60 characters.

-a tells infocmp to retain commented-out capabilities rather than discarding them. Capa? bilities are commented by prefixing them with a period.

-D tells infocmp to print the database locations that it knows about, and exit.

-E Dump the capabilities of the given terminal as tables, needed in the C initializer for a TERMTYPE structure (the terminal capability structure in the <term.h>). This option is useful for preparing versions of the curses library hardwired for a given terminal type. The tables are all declared static, and are named according to the type and the name of the corresponding terminal entry.

Before ncurses 5.0, the split between the -e and -E options was not needed; but support for extended names required making the arrays of terminal capabilities separate from the TERMTYPE structure.

-e Dump the capabilities of the given terminal as a C initializer for a TERMTYPE structure (the terminal capability structure in the <term.h>). This option is useful for prepar? ing versions of the curses library hardwired for a given terminal type.

-F compare terminfo files. This assumes that two following arguments are filenames. The files are searched for pairwise matches between entries, with two entries considered to match if any of their names do. The report printed to standard output lists entries with no matches in the other file, and entries with more than one match. For entries with exactly one match it includes a difference report. Normally, to reduce the volume of the report, use references are not resolved before looking for differences, but res? olution can be forced by also specifying -r.

-f Display complex terminfo strings which contain if/then/else/endif expressions indented for readability.

-G Display constant literals in decimal form rather than their character equivalents.

-g Display constant character literals in quoted form rather than their decimal equiva? lents.

-i Analyze the initialization (is1, is2, is3), and reset (rs1, rs2, rs3), strings in the entry, as well as those used for starting/stopping cursor-positioning mode (smcup, rm? cup) as well as starting/stopping keymap mode (smkx, rmkx).

For each string, the code tries to analyze it into actions in terms of the other capa? bilities in the entry, certain X3.64/ISO 6429/ECMA-48 capabilities, and certain DEC VT-series private modes (the set of recognized special sequences has been selected for completeness over the existing terminfo database). Each report line consists of the capability name, followed by a colon and space, followed by a printable expansion of the capability string with sections matching recognized actions translated into {}-bracketed descriptions.

Here is a list of the DEC/ANSI special sequences recognized:

Action	Meaning
??????????????????????????????????????	
RIS	full reset
SC	save cursor
RC	restore cursor
LL	home-down
RSR	reset scroll region
??????????????????????????????????????	
DECSTR	soft reset (VT320)
S7C1T	7-bit controls (VT220)
??????????????????????????????????????	
ISO DEC G0	enable DEC graphics for G0
ISO UK G0	enable UK chars for G0
ISO US G0	enable US chars for G0
ISO DEC G1	enable DEC graphics for G1
ISO UK G1	enable UK chars for G1
ISO US G1	enable US chars for G1
??????????????????????????????????????	
DECPAM	application keypad mode
DECPNM	normal keypad mode
DECANSI	enter ANSI mode
??????????????????????????????????????	
ECMA[+-]AM	keyboard action mode
ECMA[+-]IRM	insert replace mode
ECMA[+-]SRM	send receive mode

ECMA[+-]LNM linefeed mode

??????????????????????????????????????

DEC[+-]CKM application cursor keys

DEC[+-]ANM set VT52 mode

DEC[+-]COLM 132-column mode

DEC[+-]SCLM smooth scroll

DEC[+-]SCNM reverse video mode

DEC[+-]OM origin mode

DEC[+-]AWM wraparound mode

DEC[+-]ARM auto-repeat mode

It also recognizes a SGR action corresponding to ANSI/ISO 6429/ECMA Set Graphics Rendition, with the values NORMAL, BOLD, UNDERLINE, BLINK, and REVERSE. All but NORMAL may be prefixed with

? ?+? (turn on) or

? ?-? (turn off).

An SGR0 designates an empty highlight sequence (equivalent to {SGR:NORMAL}).

-I Set output format to terminfo.

-p Ignore padding specifications when comparing strings.

-Q n Rather than show source in terminfo (text) format, print the compiled (binary) format in hexadecimal or base64 form, depending on the option's value:

1 hexadecimal

2 base64

3 hexadecimal and base64

For example, this prints the compiled terminfo value as a string which could be as? signed to the TERMINFO environment variable:

infocmp -0 -q -Q2

-q This makes the output a little shorter:

? Make the comparison listing shorter by omitting subheadings, and using ?-? for ab? sent capabilities, ?@? for canceled rather than ?NULL?.

? However, show differences between absent and canceled capabilities.

? Omit the ?Reconstructed from? comment for source listings.

-Rsubset

Restrict output to a given subset. This option is for use with archaic versions of

terminfo like those on SVr1, Ultrix, or HP-UX that do not support the full set of SVR4/XSI Curses terminfo; and variants such as AIX that have their own extensions in? compatible with SVr4/XSI.

? Available terminfo subsets are ?SVr1?, ?Ultrix?, ?HP?, and ?AIX?; see terminfo(5) for details.

? You can also choose the subset ?BSD? which selects only capabilities with termcap equivalents recognized by 4.4BSD.

? If you select any other value for -R, it is the same as no subset, i.e., all capa? bilities are used.

A few options override the subset selected with -R, if they are processed later in the command parameters:

-C sets the ?BSD? subset as a side effect.

-I sets the subset to all capabilities.

-r sets the subset to all capabilities.

-s [d|i|l|c]

The -s option sorts the fields within each type according to the argument below:

d leave fields in the order that they are stored in the terminfo database.

i sort by terminfo name.

l sort by the long C variable name.

c sort by the termcap name.

If the -s option is not given, the fields printed out will be sorted alphabetically by the terminfo name within each type, except in the case of the -C or the -L options, which cause the sorting to be done by the termcap name or the long C variable name, re? spectively.

-T eliminates size-restrictions on the generated text. This is mainly useful for testing and analysis, since the compiled descriptions are limited (e.g., 1023 for termcap, 4096 for terminfo).

-t tells tic to discard commented-out capabilities. Normally when translating from ter? minfo to termcap, untranslatable capabilities are commented-out.

-U tells infocmp to not post-process the data after parsing the source file. This feature helps when comparing the actual contents of two source files, since it excludes the in? ferences that infocmp makes to fill in missing data.

-V reports the version of ncurses which was used in this program, and exits.

-v n prints out tracing information on standard error as the program runs.

The optional parameter n is a number from 1 to 10, inclusive, indicating the desired level of detail of information. If ncurses is built without tracing support, the op? tional parameter is ignored.

-W By itself, the -w option will not force long strings to be wrapped. Use the -W option to do this.

-w width

changes the output to width characters.

-x print information for user-defined capabilities (see user_caps(5). These are exten? sions to the terminfo repertoire which can be loaded using the -x option of tic.

FILES

/etc/terminfo

compiled terminal description database

EXTENSIONS

The -0, -1, -a, -e, -E, -f, -F, -g, -G, -i, -l, -p, -q, -Q, -R, -t, -T, and -V options are ncurses extensions.

PORTABILITY

X/Open Curses Issue 7 (2009) specifies infocmp. It does not mention options for producing descriptions in termcap format.

SVr4 infocmp does not distinguish between absent and canceled capabilities. It furthermore reports missing integer capabilities as -1 (its internal representation). ncurses shows these as ?NULL? for consistency with missing string capabilities.

The -r option of ncurses infocmp uses SVr4's notion of ?termcap? capabilities. BSD curses had a more restricted set. To see only those present in 4.4BSD, use ?-r -RBSD?.

HISTORY

Although System V Release 2 provided a terminfo library, it had no documented tool for de? compiling the terminal descriptions. Tony Hansen (AT&T) wrote the first infocmp in early 1984, for System V Release 3.

Eric Raymond used the AT&T documentation in 1995 to provide an equivalent infocmp for ncurses. In addition, he added a few new features such as:

? the -e option, to support fallback (compiled-in) terminal descriptions

? the -i option, to help with analysis

Later, Thomas Dickey added the -x (user-defined capabilities) option, and the -E option to

support fallback entries with user-defined capabilities.

For a complete list, see the EXTENSIONS section.

In 2010, Roy Marples provided an infocmp program for NetBSD. It is less capable than the SVr4 or ncurses versions (e.g., it lacks the sorting options documented in X/Open), but does include the -x option adapted from ncurses.

BUGS

The -F option of infocmp(1) should be a toe(1) mode.

AUTHORS

Eric S. Raymond <esr@snark.thyrsus.com> and

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

SEE ALSO

captoinfo(1), infotocap(1), tic(1), toe(1), ncurses(3NCURSES), terminfo(5), user_caps(5)

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