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

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'hexdump.1' command

$ man hexdump.1

HEXDUMP(1) User Commands HEXDUMP(1)

NAME

hexdump - display file contents in hexadecimal, decimal, octal, or ascii

hexdump options file ...

hd options file ...

DESCRIPTION

The hexdump utility is a filter which displays the specified files, or standard input if no files are specified, in a user-specified format.

OPTIONS

Below, the length and offset arguments may be followed by the multiplicative suffixes KiB (=1024), MiB (=1024*1024), and so on for GiB, TiB, PiB, EiB, ZiB and YiB (the "iB" is optional, e.g., "K" has the same meaning as "KiB"), or the suffixes KB (=1000), MB (=1000*1000), and so on for GB, TB, PB, EB, ZB and YB.

-b, --one-byte-octal

One-byte octal display. Display the input offset in hexadecimal, followed by sixteen space-separated, three-column, zero-filled bytes of input data, in octal, per line.

-c, --one-byte-char

One-byte character display. Display the input offset in hexadecimal, followed by sixteen space-separated, three-column, space-filled characters of input data per line.

-C, --canonical

Canonical hex+ASCII display. Display the input offset in hexadecimal, followed by sixteen space-separated, two-column, hexadecimal bytes, followed by the same sixteen bytes in %_p format enclosed in '|' characters. Invoking the program as hd implies this option.

-d, --two-bytes-decimal

Two-byte decimal display. Display the input offset in hexadecimal, followed by eight space-separated, five-column, zero-filled, two-byte units of input data, in unsigned decimal, per line.

-e, --format format_string

Specify a format string to be used for displaying data.

-f, --format-file file

Specify a file that contains one or more newline-separated format strings. Empty lines and lines whose first non-blank character is a hash mark (#) are ignored.

-L, --color[=when]

Accept color units for the output. The optional argument when can be auto, never or always. If the when argument is omitted, it defaults to auto. The colors can be disabled; for the current built-in default see the --help output. See also the Colors subsection and the COLORS section below.

-n, --length length

Interpret only length bytes of input.

-o, --two-bytes-octal

Two-byte octal display. Display the input offset in hexadecimal, followed by eight space-separated, six-column, zero-filled, two-byte quantities of input data, in octal, per line.

-s, --skip offset

Skip offset bytes from the beginning of the input.

-v, --no-squeezing

The -v option causes hexdump to display all input data. Without the -v option, any number of groups of output lines which would be

identical to the immediately preceding group of output lines (except for the input offsets), are replaced with a line comprised of a single asterisk.

-x, --two-bytes-hex

Two-byte hexadecimal display. Display the input offset in hexadecimal, followed by eight space-separated, four-column, zero-filled, two-byte quantities of input data, in hexadecimal, per line.

-V, --version

Display version information and exit.

-h, --help

Display help text and exit.

For each input file, hexdump sequentially copies the input to standard output, transforming the data according to the format strings specified by the -e and -f options, in the order that they were specified.

FORMATS

A format string contains any number of format units, separated by whitespace. A format unit contains up to three items: an iteration count, a byte count, and a format.

The iteration count is an optional positive integer, which defaults to one. Each format is applied iteration count times.

The byte count is an optional positive integer. If specified it defines the number of bytes to be interpreted by each iteration of the format.

If an iteration count and/or a byte count is specified, a single slash must be placed after the iteration count and/or before the byte count to disambiguate them. Any whitespace before or after the slash is ignored.

The format is required and must be surrounded by double quote (" ") marks. It is interpreted as a fprintf-style format string (see fprintf(3), with the following exceptions:

1.

An asterisk (*) may not be used as a field width or precision.

2.

A byte count or field precision is required for each s conversion character (unlike the fprintf3 default which prints the entire string if the precision is unspecified).

3. The conversion characters h, l, n, p, and q are not supported.

4. The single character escape sequences described in the C standard are supported:

?	?
?NULL	? \0 ?
?	?
?<alert character>	? \a ?
?	?
?<backspace>	? \b ?
?	?
?<form-feed>	? \f ?
?	?
?<newline>	? \n ?
?	?
?<carriage return>	? \r ?
?	?
?<tab>	? \t ?
?	?
?<vertical tab>	? \v ?
?	

Conversion strings

The hexdump utility also supports the following additional conversion strings.

_a[dox]

Display the input offset, cumulative across input files, of the next byte to be displayed. The appended characters d, o, and x specify the display base as decimal, octal or hexadecimal respectively.

_A[dox]

Identical to the _a conversion string except that it is only performed once, when all of the input data has been processed.

_c

Output characters in the default character set. Non-printing characters are displayed in three-character, zero-padded octal, except for those representable by standard escape notation (see above), which are displayed as two-character strings.

_p

Output characters in the default character set. Non-printing characters are displayed as a single '.'.

_u

Output US ASCII characters, with the exception that control characters are displayed using the following, lower-case, names. Characters greater than 0xff, hexadecimal, are displayed as hexadecimal strings.

??

? ? ? ? ? ? ?

?000 nul ? 001 soh ? 002 stx ? 003 etx ? 004 eot ? 005 enq ?

??

? ? ? ? ? ? ?

?006 ack ? 007 bel ? 008 bs ? 009 ht ? 00A lf ? 00B vt ?

??

? ? ? ? ? ? ?

?00C ff ? 00D cr ? 00E so ? 00F si ? 010 dle ? 011 dc1 ?

??

? ? ? ? ? ? ?

?012 dc2 ? 013 dc3 ? 014 dc4 ? 015 nak ? 016 syn ? 017 etb ?

??

? ? ? ? ? ? ?

?018 can ? 019 em ? 01A sub ? 01B esc ? 01C fs ? 01D gs ?

??

? ? ? ? ? ? ?

?01E rs ? 01F us ? 0FF del ? ? ? ?

??

Colors

When put at the end of a format specifier, hexdump highlights the respective string with the color specified. Conditions, if present, are evaluated prior to highlighting.

_L[color_unit_1,color_unit_2,...,color_unit_n]

The full syntax of a color unit is as follows:

[!]COLOR[:VALUE][@OFFSET_START[-END]]

!

Negate the condition. Please note that it only makes sense to negate a unit if both a value/string and an offset are specified. In that case the respective output string will be highlighted if and only if the value/string does not match the one at the offset.

COLOR

One of the 8 basic shell colors.

VALUE

A value to be matched specified in hexadecimal, or octal base, or as a string. Please note that the usual C escape sequences are not interpreted by hexdump inside the color_units.

OFFSET

An offset or an offset range at which to check for a match. Please note that lone OFFSET_START uses the same value as END offset.

Counters

The default and supported byte counts for the conversion characters are as follows:

%_c, %_p, %_u, %c

One byte counts only.

%d, %i, %o, %u, %X, %x

Four byte default, one, two and four byte counts supported.

%E, %e, %f, %G, %g

Eight byte default, four byte counts supported.

The amount of data interpreted by each format string is the sum of the data required by each format unit, which is the iteration count times the byte count, or the iteration count times the number of bytes required by the format if the byte count is not specified.

The input is manipulated in blocks, where a block is defined as the largest amount of data specified by any format string. Format strings interpreting less than an input block?s worth of data, whose last format unit both interprets some number of bytes and does not have a specified iteration count, have the iteration count incremented until the entire input block has been processed or there is not enough data remaining in the block to satisfy the format string.

If, either as a result of user specification or hexdump modifying the iteration count as described above, an iteration count is greater than one, no trailing whitespace characters are output during the last iteration.

It is an error to specify a byte count as well as multiple

conversion characters or strings unless all but one of the conversion characters or strings is _a or _A.

If, as a result of the specification of the -n option or end-of-file being reached, input data only partially satisfies a format string, the input block is zero-padded sufficiently to display all available data (i.e., any format units overlapping the end of data will display some number of the zero bytes).

Further output by such format strings is replaced by an equivalent number of spaces. An equivalent number of spaces is defined as the number of spaces output by an s conversion character with the same field width and precision as the original conversion character or conversion string but with any '+', ' ', '#' conversion flag characters removed, and referencing a NULL string.

If no format strings are specified, the default display is very similar to the -x output format (the -x option causes more space to be used between format units than in the default output).

EXIT STATUS

hexdump exits 0 on success and > 0 if an error occurred.

CONFORMING TO

The hexdump utility is expected to be IEEE Std 1003.2 ("POSIX.2") compatible.

EXAMPLES

Display the input in perusal format:

```
"%06.6_ao "  12/1 "%3_u "
"\t" "%_p "
"\n"
```

Implement the -x option:

```
"%07.7_Ax\n"
"%07.7_ax  " 8/2 "%04x " "\n"
```

MBR Boot Signature example: Highlight the addresses cyan

and the bytes at offsets 510 and 511 green if their value

is 0xAA55, red otherwise.

"%07.7_Ax_L[cyan]\n"

"%07.7_ax_L[cyan] " 8/2 "   %04x_L[green:0xAA55@510-511,!red:0xAA55@510-511] " "\n"

COLORS

Implicit coloring can be disabled by an empty file

/etc/terminal-colors.d/hexdump.disable.

See terminal-colors.d(5) for more details about

colorization configuration.

REPORTING BUGS

For bug reports, use the issue tracker at

https://github.com/karelzak/util-linux/issues.

AVAILABILITY

The hexdump command is part of the util-linux package which

can be downloaded from Linux Kernel Archive

<https://www.kernel.org/pub/linux/utils/util-linux/>.