NAME

xxd − make a hexdump or do the reverse.

SYNOPSIS

xxd −h[elp]
xxd [options] [infile [outfile]]
xxd −r[evert] [options] [infile [outfile]]

DESCRIPTION

xxd creates a hex dump of a given file or standard input. It can also convert a hex dump back to its original binary form. Like **uuencode**(1) and **uudecode**(1) it allows the transmission of binary data in a 'mail-safe' ASCII representation, but has the advantage of decoding to standard output. Moreover, it can be used to perform binary file patching.

OPTIONS

If no *infile* is given, standard input is read. If *infile* is specified as a `**−**' character, then input is taken from standard input. If no *outfile* is given (or a `**−**' character is in its place), results are sent to standard output.

Note that a "lazy" parser is used which does not check for more than the first option letter, unless the option is followed by a parameter. Spaces between a single option letter and its parameter are optional. Parameters to options can be specified in decimal, hexadecimal or octal notation. Thus **−c8**, **−c 8**, **−c 010** and **−cols 8** are all equivalent.

−a | *−autoskip*
: Toggle autoskip: A single '*' replaces nul-lines. Default off.

−b | *−bits*
: Switch to bits (binary digits) dump, rather than hexdump. This option writes octets as eight digits "1"s and "0"s instead of a normal hexadecimal dump. Each line is preceded by a line number in hexadecimal and followed by an ascii (or ebcdic) representation. The command line switches −r, −p, −i do not work with this mode.

−c cols | *−cols cols*
: Format <*cols*> octets per line. Default 16 (−i: 12, −ps: 30, −b: 6). Max 256.

−C | *−capitalize*
: Capitalize variable names in C include file style, when using −i.

−E | *−EBCDIC*
: Change the character encoding in the righthand column from ASCII to EBCDIC. This does not change the hexadecimal representation. The option is meaningless in combinations with −r, −p or −i.

−e
: Switch to little-endian hexdump. This option treats byte groups as words in little-endian byte order. The default grouping of 4 bytes may be changed using *−g*. This option only applies to hexdump, leaving the ASCII (or EBCDIC) representation unchanged. The command line switches −r, −p, −i do not work with this mode.

−g bytes | *−groupsize bytes*
: Separate the output of every <*bytes*> bytes (two hex characters or eight bit-digits each) by a whitespace. Specify *−g 0* to suppress grouping. <*Bytes*> defaults to *2* in normal mode, *4* in little-endian mode and *1* in bits mode. Grouping does not apply to postscript or include style.

−h | *−help*
: Print a summary of available commands and exit. No hex dumping is performed.

−i | *−include*
: Output in C include file style. A complete static array definition is written (named after the input file), unless xxd reads from stdin.

−l len | *−len len*
: Stop after writing <*len*> octets.

−o offset
Add <*offset*> to the displayed file position.

−p | −ps | −postscript | −plain
Output in postscript continuous hexdump style. Also known as plain hexdump style.

−r | −revert
Reverse operation: convert (or patch) hexdump into binary. If not writing to stdout, xxd writes into its output file without truncating it. Use the combination *−r −p* to read plain hexadecimal dumps without line number information and without a particular column layout. Additional White-space and line-breaks are allowed anywhere.

−seek offset
When used after *−r*: revert with <*offset*> added to file positions found in hexdump.

−s [+][−]seek
Start at <*seek*> bytes abs. (or rel.) infile offset. + indicates that the seek is relative to the current stdin file position (meaningless when not reading from stdin). − indicates that the seek should be that many characters from the end of the input (or if combined with +: before the current stdin file position). Without −s option, xxd starts at the current file position.

−u Use upper case hex letters. Default is lower case.

−v | −version
Show version string.

CAVEATS

xxd −r has some builtin magic while evaluating line number information. If the output file is seekable, then the linenumbers at the start of each hexdump line may be out of order, lines may be missing, or overlapping. In these cases xxd will lseek(2) to the next position. If the output file is not seekable, only gaps are allowed, which will be filled by null-bytes.

xxd −r never generates parse errors. Garbage is silently skipped.

When editing hexdumps, please note that *xxd −r* skips everything on the input line after reading enough columns of hexadecimal data (see option −c). This also means, that changes to the printable ascii (or ebcdic) columns are always ignored. Reverting a plain (or postscript) style hexdump with xxd −r −p does not depend on the correct number of columns. Here anything that looks like a pair of hex-digits is interpreted.

Note the difference between
% xxd −i file
and
% xxd −i < file

xxd −s +seek may be different from *xxd −s seek*, as lseek(2) is used to "rewind" input. A '+' makes a difference if the input source is stdin, and if stdin's file position is not at the start of the file by the time xxd is started and given its input. The following examples may help to clarify (or further confuse!)...

Rewind stdin before reading; needed because the 'cat' has already read to the end of stdin.
% sh −c "cat > plain_copy; xxd −s 0 > hex_copy" < file

Hexdump from file position 0x480 (=1024+128) onwards. The '+' sign means "relative to the current position", thus the '128' adds to the 1k where dd left off.
% sh −c "dd of=plain_snippet bs=1k count=1; xxd −s +128 > hex_snippet" < file

Hexdump from file position 0x100 (= 1024−768) on.
% sh −c "dd of=plain_snippet bs=1k count=1; xxd −s +−768 > hex_snippet" < file

However, this is a rare situation and the use of '+' is rarely needed. The author prefers to monitor the effect of xxd with strace(1) or truss(1), whenever −s is used.

EXAMPLES

Print everything but the first three lines (hex 0x30 bytes) of **file**.
% xxd −s 0x30 file

Print 3 lines (hex 0x30 bytes) from the end of **file**.
% xxd −s −0x30 file

Print 120 bytes as continuous hexdump with 20 octets per line.
% xxd −l 120 −ps −c 20 xxd.1
2e5448205858442031202241756775737420313 9
39362220224d616e75616c207061676520666f72
20787864220a2e5c220a2e5c222032317374204d
617920313939360a2e5c22204d616e2070616765
20617574686f723a0a2e5c2220202020546f6e79
204e7567656e74203c746f6e79407363746e7567

Hexdump the first 120 bytes of this man page with 12 octets per line.
% xxd −l 120 −c 12 xxd.1
0000000: 2e54 4820 5858 4420 3120 2241 .TH XXD 1 "A
000000c: 7567 7573 7420 3139 3936 2220 ugust 1996"
0000018: 224d 616e 7561 6c20 7061 6765 "Manual page
0000024: 2066 6f72 2078 7864 220a 2e5c for xxd"..\
0000030: 220a 2e5c 2220 3231 7374 204d "..\" 21st M
000003c: 6179 2031 3939 360a 2e5c 2220 ay 1996..\"
0000048: 4d61 6e20 7061 6765 2061 7574 Man page aut
0000054: 686f 723a 0a2e 5c22 2020 2020 hor:..\"
0000060: 546f 6e79 204e 7567 656e 7420 Tony Nugent
000006c: 3c74 6f6e 7940 7363 746e 7567 <tony@sctnug

Display just the date from the file xxd.1
% xxd −s 0x36 −l 13 −c 13 xxd.1
0000036: 3231 7374 204d 6179 2031 3939 36 21st May 1996

Copy **input_file** to **output_file** and prepend 100 bytes of value 0x00.
% xxd input_file | xxd −r −s 100 > output_file

Patch the date in the file xxd.1
% echo "0000037: 3574 68" | xxd −r − xxd.1
% xxd −s 0x36 −l 13 −c 13 xxd.1
0000036: 3235 7468 204d 6179 2031 3939 36 25th May 1996

Create a 65537 byte file with all bytes 0x00, except for the last one which is 'A' (hex 0x41).
% echo "010000: 41" | xxd −r > file

Hexdump this file with autoskip.
% xxd −a −c 12 file
0000000: 0000 0000 0000 0000 0000 0000 ....
*
000fffc: 0000 0000 40 ....A

Create a 1 byte file containing a single 'A' character. The number after '−r −s' adds to the linenumbers found in the file; in effect, the leading bytes are suppressed.
% echo "010000: 41" | xxd −r −s −0x10000 > file

Use xxd as a filter within an editor such as **vim(1)** to hexdump a region marked between 'a' and 'z'.
:'a,'z!xxd

Use xxd as a filter within an editor such as **vim(1)** to recover a binary hexdump marked between 'a' and 'z'.
:'a,'z!xxd −r

Use xxd as a filter within an editor such as **vim(1)** to recover one line of a hexdump. Move the cursor over the line and type:
!!xxd −r

Read single characters from a serial line
% xxd −c1 < /dev/term/b &
% stty < /dev/term/b −echo −opost −isig −icanon min 1
% echo −n foo > /dev/term/b

RETURN VALUES

The following error values are returned:

0	no errors encountered.
−1	operation not supported (*xxd −r −i* still impossible).
1	error while parsing options.
2	problems with input file.
3	problems with output file.
4,5	desired seek position is unreachable.

SEE ALSO

uuencode(1), uudecode(1), patch(1)

WARNINGS

The tools weirdness matches its creators brain. Use entirely at your own risk. Copy files. Trace it. Become a wizard.

VERSION

This manual page documents xxd version 1.7

AUTHOR

(c) 1990-1997 by Juergen Weigert
<jnweiger@informatik.uni−erlangen.de>

Distribute freely and credit me,
make money and share with me,
lose money and don't ask me.

Manual page started by Tony Nugent
<tony@sctnugen.ppp.gu.edu.au> <T.Nugent@sct.gu.edu.au>
Small changes by Bram Moolenaar. Edited by Juergen Weigert.