

**NAME**

hex2bin – Create a binary from its hexdump.

**SYNOPSIS**

**hex2bin** [-q|-f|-v|-I] [[-i] *inputfile*] [-o *outputfile*]

**DESCRIPTION**

*hex2bin* is the counterpart to *hexdump*.

Together with *hexdump* it can be used for editing binary files: First you *hexdump* the file, then you modify the dump-file in an editor, then you use *hex2bin* to produce the modified binary file. Because there are so many versions of *hexdump* out there, which have so many different output-formats, you had better use that one distributed together with *hex2bin*.

Also, you can use this program to produce binary data to send to a device or to another program. Use option **-I** for this task.

**-i** *inputfile* || *inputfile*

At most one inputfile can be given. If none is, then stdin will be used for input. (Giving more inputfiles would make no sense.)

**-o** *outputfile*

At most one outputfile can be given. If none is, then stdout will be used for output. On systems where textfiles and binaries are treated differently (such as MS-DOS) an outputfile should be given, for output-redirection may be flakey with binary data.

**-q** (quiet) hush up warnings. Errors are still printed.

**-f** (flush) flush output after each chunk of input.

**-v** (verbose) print more infos.

**-I** (interactive) Imply **-f** and **-v** and prompt current offset.

**INPUTFILE-SYNTAX**

Inputlines usually consist of an **address**-field terminated by a colon (:) followed by byte-**data** or strings. Anything after a pipe-character (|) is ignored (Comments).

Each, any or all of these parts can be omitted.

A special case are **repetition**-lines, which start with an asterisk (\*) and may be followed by a decimal(base 10) number.

The last line consists of a semicolon (;) optionally preceded by an Address-field. Anything after this semicolon will be ignored. If no line with semicolon is found till the end of input and option **-q** is not specified, a warning is shown, but all data-fields are still converted. The reason for issuing the warning at all is that *hexdump* outputs a semicolon-line at the end, and thus the warning most likely indicates unexpected truncation. In interactive mode (option: **-I**), this warning is suppressed.

**Address-field:**

A hexadecimal number, whose actual value is important ONLY before and after repetition-lines. If the value does not match the internal data-counting, then a warning message is printed (unless option **-q** is specified) and a variable is set to the difference, which will then be added to all subsequent addresses. Thus if you insert or delete data (or lines) in the dump-file, there is no need to change all further addresses.

If the first (hex-)number of an input-line is not followed by a colon or semicolon, it will be interpreted as part of the data-field.

**Data-field:**

can be a sequence of hexadecimal values separated by spaces mixed with strings, which are handled c-like (escape-codes \n,\a,\nnn,...). Thus the following is a valid Data-field:

```
0A 7f " \aBeep\t" - 04 1b "\0"
```

which will be converted to a newline(0A), a Delete-char(7F), a space (first character of string), a backslash(\), a beep-code(\a), the characters 'B' 'e' 'e' 'p', a tab, an EOF-code, an Esc-code and the Null-byte. Dashes ('-') will be ignored for compatibility with hexdump's output.

**Repetition-lines:**

Lines that begin with an asterisk will cause the latest data-field to be repeated a certain number of times. There must have been at least one line containing at least one byte of data before the first repetition line, or an error message is issued and conversion aborted.

The actual number of repetitions can be specified in two ways:

**explicitly:** if a **decimal** number follows the asterisk, this number will specify the number of repetitions. (Note: to get a total of e.g. 10 times a chunk, specify 9, because the chunk itself was already output once where it appeared in the input.)

**implicitly:** If no explicit count is specified, the number of repetitions is automatically calculated based on the address-field of the next input line. In this case, an address **must** be specified there, otherwise an error occurs and conversion is aborted. Even more, if the number of bytes necessary to reach the next specified address is not a multiple of the byte length of the repeated chunk, this is also an error.

**BUGS**

none ;-)

Report missing features (in your eyes) to author's address

**SEE ALSO**

hexdump(1)

**COPYRIGHT**

Copyright (c) 1994 by Andreas Leitgeb (AvL)  
(avl@logic.at)

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

**NAME**

hexdump – reversible hexdump

**SYNOPSIS**

**hexdump** [ **-n** ] [ **-f** ] [ [ **-i** ] *inputfile* ] [ **-o** *outputfile* ]

**DESCRIPTION**

*hexdump* is the standard utility for viewing binary files.

Together with *hex2bin*, *hexdump* is also a utility for editing binary files. First you *hexdump* the file, then you modify the dump-file in an editor, then you use *hex2bin* to produce the modified binary file.

**OPTIONS**

- n** causes hexdump to display all input data. Without this option, any number of output lines that are identical to the immediately preceding output line (except for the input offset), are replaced with a line comprised of a single asterisk. This makes output more concise e.g. if input contains a long sequence of one repeated byte value.
- f** (flush-mode) causes hexdump to reflect its input immediately. Without this option, output is buffered (on ttys this means linebuffered, otherwise even blockbuffered). This option is especially useful, if you want to watch bytes as they are trickling out of a binary device. (e.g. /dev/mouse, /dev/ttyS1)

For obvious reasons, this implies **-n**.

[**-i**] *inputfile*

at most one inputfile can be given. if none is, then stdin will be used for input. (Giving more Inputfiles would have no sense.) On systems where textfiles and binaries are treated differently (such as MS-DOS) an inputfile should be given, as input-redirection may be flakey with binary data.

**-o** *outputfile*

at most one outputfile can be given. if none is, then stdout will be used for output.

**BUGS**

none ;-)

**SEE ALSO**

*hex2bin*(1)

**COPYRIGHT**

Copyright (c) 2001 by Andreas Leitgeb (AvL)  
(avl@logic.at)

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.