# hexdump examples

The hexdump man page is a bit frustrating, but some better help can be found.

Two tips for reading the man page:

- “+o” is just a list bullet, not a conversion character
- the first few “EXAMPLES” require the -f option

Here are a few more examples to jump-start new users…

The input file (hexdump.txt)

```
The correction for the aberration of light is said,
on high authority, not to be perfect even in that most perfect organ, the
eye.
```

Decimal byte offsets, 16 bytes per line

1. “Print the current decimal byte offset padded to 10 digits with 0's, then print the | pipe character. Next take 16 items each 1 byte in size and print each item as a printing character. Finally print another | pipe character followed by a newline.”

```
hexdump -v -e '"%010_ad  |" 16/1 "%_p" "|\n"' hexdump.txt
```

```
0000000000  |The correction f|
0000000016  |or the aberratio|
0000000032  |n of light is sa|
0000000048  |id,.on high auth|
0000000064  |ority, not to be|
0000000080  | perfect even in|
0000000096  | that most perfe|
0000000112  |ct organ, the.ey|
0000000128  |e..|
```

The same thing, broken up into two expressions

1. “Take 1 item of 16 bytes and print the current decimal byte offset padded to 10 digits with 0's, then print the | pipe character.”
2. “Also take 16 items each 1 byte in size and print each item as a printing character. Then print a | pipe character followed by a newline.”

Note that the starting byte offset only changes after both expressions have been evaluated, so the 16 printing characters still start at offset 0.

```
hexdump -v -e '/16 "%010_ad  |"' -e '16/1 "%_p" "|\n"' hexdump.txt
```

```
0000000000  |The correction f|
0000000016  |or the aberratio|
0000000032  |n of light is sa|
0000000048  |id,.on high auth|
0000000064  |ority, not to be|
0000000080  | perfect even in|
0000000096  | that most perfe|
0000000112  |ct organ, the.ey|
0000000128  |e..|
```

The same thing once more, broken up into three expressions

1. “Take 1 item of 16 bytes and print the current decimal byte offset padded to 10 digits with 0's, then print the | pipe character.”
2. “Take those same 16 bytes again, but this time divide them into 1-byte items, and print each item as a printing character.”
3. “Take the same 16 byes once more, go back to counting them as just 1 item of 16 bytes, and print a | pipe character followed by a newline.”

```
hexdump -v -e '/16 "%010_ad  |"' -e '/1 "%_p"' -e '/16 "|\n"' hexdump.txt
```

```
0000000000  |The correction f|
0000000016  |or the aberratio|
0000000032  |n of light is sa|
0000000048  |id,.on high auth|
0000000064  |ority, not to be|
0000000080  | perfect even in|
0000000096  | that most perfe|
0000000112  |ct organ, the.ey|
0000000128  |e..|
```

Again as three expressions, but using only one iteration/byte count

```
hexdump -v -e '/16 "%010_ad  |"' -e '"%_p"' -e '"|\n"' hexdump.txt
```

```
hexdump -v -e '"%010_ad  |"' -e '16/1 "%_p"' -e '"|\n"' hexdump.txt
```

```
hexdump -v -e '"%010_ad  |"' -e '"%_p"' -e '/16 "|\n"' hexdump.txt
```

All three of these are identical because of the way `hexdump` decides the size of the “block” (term from the `man` page) to use when reading through the file.

Offset printed for each byte, left-aligned characters, 7 bytes per line

```
hexdump -v -e '7/1 "%5_ad:%-5_c" "\n"' hexdump.txt
```

```
    0:T        1:h        2:e        3:         4:c        5:o        6:r
    7:r        8:e        9:c       10:t       11:i       12:o       13:n
   14:        15:f       16:o       17:r       18:        19:t       20:h
   21:e       22:        23:a       24:b       25:e       26:r       27:r
   28:a       29:t       30:i       31:o       32:n       33:        34:o
   35:f       36:        37:l       38:i       39:g       40:h       41:t
   42:        43:i       44:s       45:        46:s       47:a       48:i
   49:d       50:,       51:\n      52:o       53:n       54:        55:h
   56:i       57:g       58:h       59:        60:a       61:u       62:t
   63:h       64:o       65:r       66:i       67:t       68:y       69:,
   70:        71:n       72:o       73:t       74:        75:t       76:o
   77:        78:b       79:e       80:        81:p       82:e       83:r
   84:f       85:e       86:c       87:t       88:        89:e       90:v
   91:e       92:n       93:        94:i       95:n       96:        97:t
   98:h       99:a      100:t      101:       102:m      103:o      104:s
  105:t      106:       107:p      108:e      109:r      110:f      111:e
  112:c      113:t      114:       115:o      116:r      117:g      118:a
  119:n      120:,      121:       122:t      123:h      124:e      125:\n
  126:e      127:y      128:e      129:.      130:\n          :          :
```

The effect of `-v`

With `-v`

```
( echo "abcdefghijklmno"; echo "abcdefghijklmno" ) | hexdump -v -C
```

```
00000000  61 62 63 64 65 66 67 68  69 6a 6b 6c 6d 6e 6f 0a  |abcdefghijklmno.|
00000010  61 62 63 64 65 66 67 68  69 6a 6b 6c 6d 6e 6f 0a  |abcdefghijklmno.|
00000020
```

Without `-v`

```
( echo "abcdefghijklmno"; echo "abcdefghijklmno" ) | hexdump -C
```

```
00000000  61 62 63 64 65 66 67 68  69 6a 6b 6c 6d 6e 6f 0a  |abcdefghijklmno.|
*
00000020
```

One `hexdump` vs. `od` example

```
hexdump  -e ' "%07_ad" 16/1 "  %2_c" "\n"' hexdump.txt
```

```
0000000   T   h   e       c   o   r   r   e   c   t   i   o   n       f
0000016   o   r       t   h   e       a   b   e   r   r   a   t   i   o
0000032   n       o   f       l   i   g   h   t       i   s       s   a
0000048   i   d   ,  \n   o   n       h   i   g   h       a   u   t   h
0000064   o   r   i   t   y   ,       n   o   t       t   o       b   e
0000080       p   e   r   f   e   c   t       e   v   e   n       i   n
0000096       t   h   a   t       m   o   s   t       p   e   r   f   e
0000112   c   t       o   r   g   a   n   ,       t   h   e  \n   e   y
0000128   e   .  \n
```

```
od -Ad -w16 -tc hexdump.txt
```

```
0000000   T   h   e       c   o   r   r   e   c   t   i   o   n       f
0000016   o   r       t   h   e       a   b   e   r   r   a   t   i   o
0000032   n       o   f       l   i   g   h   t       i   s       s   a
0000048   i   d   ,  \n   o   n       h   i   g   h       a   u   t   h
0000064   o   r   i   t   y   ,       n   o   t       t   o       b   e
0000080       p   e   r   f   e   c   t       e   v   e   n       i   n
0000096       t   h   a   t       m   o   s   t       p   e   r   f   e
0000112   c   t       o   r   g   a   n   ,       t   h   e  \n   e   y
0000128   e   .  \n
0000131
```

Links

Manual on How to Use the Hexdump Unix Utility

> More hexdump explanation and examples.

Debian hexdump man page

> The Debian version of the hexdump man page has a few additions compared to the FreeBSD version.

FreeBSD hexdump man page

> The upstream version of the hexdump man page.

od invocation

> The od documentation from the GNU coreutils manual.

od man page

> The Debian od man page.

Found a mistake?

Submit a comment or correction

Updates

| | |
|---|---|
| 2013 Aug 17 | Explanation rewording for the three-expression example |
| 2013 Jan 08 | Comments link |
| 2012 Feb 04 | more explanation, block size examples, od example, links section |
| 2010 Dec 01 | single-expression example and some rewording |
| 2010 Nov 13 | Fixed more missing \n escapes, hopefully fixed the root problem |
| 2010 Oct 06 | Fixed missing \n newline escapes |
| 2010 Sep 13 | Initial post |