

NAME

bbe – binary block editor

SYNOPSIS

bbe [*options*]...

DESCRIPTION

bbe is a sed-like editor for binary files. It performs binary transformations on the blocks of input stream.

OPTIONS

bbe accepts the following options:

-b, --block=BLOCK

Block definition.

-e, --expression=COMMAND

Add the COMMAND to the commands to be executed.

-f, --file=script-file

Add the content of script-file to commands.

-o, --output=name

Write output to *name* instead of standard output.

-s, --suppress

Suppress normal output, print only block contents.

-, --help

List all available options and their meanings.

-V, --version

Show version of program.

BLOCK can be defined as:

N:M Where *N*'th byte starts a *M* bytes long block (first byte is 0).

:M Block length in input stream is *M*.

/start/:M

String *start* starts *M* bytes long block.

/start:/stop/

String *start* starts the block and block ends to string *stop*.

/start/: String *start* starts the block and block will end at next occurrence of *start*. Only the first *start* is included to the block.

:/stop/ Block starts at the beginning of input stream (or at the end of previous block) and ends at the next occurrence of *stop*. String *stop* will be included to the block.

Special value '\$' of *M* means the end of stream.

Default value for block is 0:\$, meaning the whole input stream.

Both *start* and *stop* strings are included to block. Nonprintable characters can be escaped as

\nnn decimal

\xnn hexadecimal

\0nnn octal

Character '\' can be escaped as '\\'. Escape codes 'a', 'b', 't', 'n', 'v', 'f', 'r' and ';' can also be used.

Length (*N* and *M*) can be defined as decimal (n), hexadecimal (xn) or octal (0n) value.

COMMAND SYNOPSIS

bbe has two type of commands: block and byte commands, both are allways related to current block. That means that the input stream outside of block remains untouched.

Block commands

- D** [*n*] Delete the *n*'th block. Without *n*, all found blocks are deleted from the output stream.
- I** *string* Insert the string *string* before the block.
- A** *string*
Append the string *string* at the end of block.
- J** *n* Skip *n* blocks before executing commands after this command.
- L** *n* Leave all blocks unmodified starting from block number *n*. Affects only commands after this command.
- N** Before printing a block, the file name in which the block starts is printed.
- F** *f* Before printing a block, the input stream offset at the begining of the block is printed. *f* can be H, D or O for Hexadecimal, Decimal or Octal format of offset.
- B** *f* Before printing a block, the block number is printed (first block == 1) *f* can be H, D or O for Hexadecimal, Decimal or Octal format of block number.
- >** *file* Before printing a block, the contents of file *file* is printed.
- <** *file* After printing a block, the contents of file *file* is printed.

Byte commands

- n* in byte commands is offset from the beginning of current block (starts from zero).
- r** *n string*
Replace bytes starting at position *n* with string *string*.
- i** *n string*
Insert *string* starting at position *n*.
- p** *format*
The contents of block is printed in format defined by *format*. *format* can have any of the formats H, D, O, A and B for Hexadecimal, Decimal, Octal, Asciii and Binary.
- s/search/replace/**
Replace all occurrences of *search* with *replace*.
- y/source/dest/**
Translate bytes in *source* to the corresponding bytes in *dest*. *Source* and *dest* must have equal length.
- d** *n m* | ***** Delete *m* bytes starting from the offset *n*. If ***** is defined instead of *m*, then all bytes starting from *n* are deleted.
- c** *from to*
Convert bytes from format *from* to *to*. Currently supported formats are:
BCD Binary coded decimal
ASC Ascii
- j** *n* Commands after the **j**-command are ignored for first *n* bytes of the block.
- l** *n* Commands after the **l**-command are ignored from *n*'th byte of the block.
- w** *file* Write bytes from the current block to file *file*. Commands before **w**-command have effect to what will be written. %B or %nB in *file* will be replaced by current block number. *n* in %nB is field length, leading zero in *n* causes the block number to be left padded with zeroes.

& c Performs binary and with *c*.
| c Performs binary or with *c*.
^ c Performs binary xor with *c*.
~ Performs binary negation.
u n c All bytes from start of the block to offset *n* are replaced by *c*.
f n c All bytes starting from offset *n* to end of the block are replaced by *c*.
x Exchange the contents of nibbles (half an octet) of bytes.

Nonvisible characters in strings can be escaped same way as in block definition strings. Character `'/'` in `s` and `y` commands can be any visible character.

Note that the `D`, `A`, `I`, `F`, `B`, `c`, `s`, `i`, `y`, `p`, `<`, `>` and `d` commands cause the length of input and output streams to be different.

EXAMPLES

`bbe -e "s/c:\\temp\\data1.txt/c:\\temp\\data2.txt/" file1`
 all occurrences of `"c:\\temp\\data1.txt"` in file `file1` are changed to `"c:\\temp\\data2.txt"`

`bbe -b 0420:16 -e "r 4 \\x12\\x4a" file1`
 Two bytes starting at fifth byte of a 16 byte long block starting at offset 0420 (octal) in `file1` are changed to hexadecimal values 12 and 4a.

`bbe -b :16 -e "A \\x0a" file1`
 Newline is added after every block, block length is 16.

SEE ALSO

`sed(1)`.

AUTHOR

Timo Savinen <tjsa@iki.fi>