MIPS SDE ELF Conversion Tool

`sde-conv`
Version 5.0

October 2003

MIPS Technologies Inc

MIPS Technologies Inc
Revision: 1.4
TEXinfo 2003-05-04.08

1 conv

```
sde-conv [ -l ] [ -v ] [ -y ] [ -p ] [ -r ]
         [ -a addr[/size] ]
         [ -a addr[:limit] ]
         [ -f fmt ] [ -R offs ]
         [ -w width ] [ -b byte,... ]
         [ -i offs ] [ -s size ]
         [ -x seg,... ] [ -o ofile ] ifile
```

The `sde-conv` utility takes ELF executable *ifile* and converts it into an ASCII or encoded binary format, suitable for downloading onto a PROM programmer or evaluation board. It can take horizontal and/or vertical slices through the file, to cater for PROM programmers that do not have this facility or memory capacity.

`-o` *`ofile`*	Selects *ofile* as the output file, the default is the standard output. Note: you must use this for the binary output format.
`-l`	Lists the segments in the ELF file's program header, for possible use in the `-x` option below; does not produce any other output.
`-v`	Selects verbose mode, which makes the program report its actions as it converts the file.
`-y`	Requests symbol output: for formats which support this the file's symbol table is also converted and sent to the output file.

`-a` *`addr`*
`-a` *`addr`*`:`*`limit`*
`-a` *`addr`*`/`*`size`*
	Selects PROM output mode and specifies the physical address range of the PROM. All input data located between *addr* and *limit* (or between *addr* and *addr+size*) are deemed to be PROM-resident, and are mapped to their relative offset from *addr*. Segments outside of this address range are assumed to be intended for relocation to RAM, and they are concatenated to the end of the last PROM-resident segment (rounded to the next 16 byte boundary). The default value for *addr* is `1fc00000`, and the default *size* is `400000` (i.e. 4Mb).
	Note that if the code segment has a load address that doesn't fall inside the ROM then it will get treated like an initialised data segment. It will get mapped to the start of the ROM (the reset exception address) if it was linked with the lowest address, and then your reset code can be carefully written to copy the code segment from ROM to RAM.

`-p`	Selects PROM output mode, equivalent to '`-a 1fc00000/400000`'.
`-r`	Selects raw binary input mode, which ignores any ELF headers and converts the whole input file into the output format, starting at the selected PROM base address.
`-R` *`offs`*	Selects raw binary input mode as above, but places the input at offset *offs* from the PROM base address.
`-f` *`fmt`*	Selects the output format, from the list below. If not specified then the environment variable `DLTYPE` is used. The default if neither of these are specified is `srec`.

`srec` `s3`	Motorola S-records, using S3 and S7 (32-bit address) records. Symbol output is provided via S4 records, as supported by LSI PMON, if the `-y` option is also used.
`s2`	Same as above, except using S2 and S8 (24-bit address) records, to suit some PROM programmers.
`s1`	Same as above, except using S1 and S9 (16-bit address) records.
`lsi`	High density (radix 64) ASCII encoding for downloading by the '`LSI PMON`' monitor; symbol output is supported with the `-y` option.
`idt`	Binary S-record format for downloading by the '`IDT/sim`' monitor.
`stagbin` `sbin`	Proprietary binary format for '`Stag`' PROM programmers.
`fl`	MIPS flash file format, suitable for downloading to a Malta or SEAD-2 board.
`bin`	Raw binary output, suitable for further conversion by another user program.
`relf`	Relocated ELF executable object file output, with the code and data relocated to their new addresses in ROM, as selected by the '`-p`' or '`-a`' options. This can be used to generate a ROM image which can be loaded by gdb's MDI interface. Note that symbols and other debug information are not copied to the new file, and the file's entrypoint address is set to the base ROM address.

`-w` *`width`* Specifies the width of CPU data bus in bits (default: 32).

`-b` *`byte,...`*

The input file is read in multiples of *width* bits, and then these bytes are selected from it. The default is '`0,1,2,3`', which sends all bytes in their file order. To perform a byte swap specify '`3,2,1,0`'. To program four PROMs with consecutive bytes, run `conv` four times with '`-b0`', '`-b1`', '`-b2`' and '`-b3`'.

To handle interleaved PROMs, multiply the PROM width by the interleave, (e.g. '`-w64`' for two-way interleave on a 32-bit bus), and then run `conv` interleave times byte-width times (e.g. '`-b0`' to '`-b7`').

`-i` *`offs`*
`-s` *`size`* Specified together to split a large program into multiple PROMs. The `-i` flag specifies the initial offset, and `-s` the size of the PROM. For example for two 128K proms:

```
conv -f s3 -i 0x00000 -s 0x20000 -o prom.s3.1 prom
conv -f s3 -i 0x20000 -s 0x20000 -o prom.s3.2 prom
```

`-x` *`seg,...`*

Exclude the following segments from the output; the segment numbers can be determined using the `-l` option described above.