NAME

p2hex - convert code files into hex files

SYNTAX

p2hex [option(s)] <name(s)> [further options/names]

DESCRIPTION

P2HEX is a tool to convert the contents of one or several code files generated by AS into HEX files. A HEX file is a common method of representing binary data in a way that is human-readable and transferrable over non-transparent data lines. Generally spoken, each byte of code or data is represented by two characters that show the byte in its hexadecimal notation. A HEX file also contains additional information like addresses and checksums that ease processing of the data. Unfortunately, there is no generally accepted standard format for HEX files. Instead, every processor manufacturer developed his own format and some of them have become "industry standards". P2HEX supports all formats that seem to have gained acceptance, with some variations that are commonplace.

Arguments to P2HEX may be either command line parameters or file name specifications. Any argument that starts with the charactes +, - or / is regarded as a comand line parameter (which may take an additional command line argument); any other argument is regarded as a file name. Generally, P2HEX needs at least two file names: An input code file and the name of the HEX output file. If multiple file names are given, P2HEX will always take the last name as the output file's name. If an input file name does not have an extension, the extension '.p' is added automatically. Similarly, the extension '.hex' is added automatically to the target file's name. A special case occurs when only one file name is given: P2HEX will then take its name as the source (possibly extended with '.p'), and the same name as target (with '.hex' as additional or replaced extension).

By default, P2HEX will choose a HEX format that is the most common for the processor family a source file contains code for; this however means that if the source file(s) contain(s) code for different processor families, the HEX file might become an undesirable mixture of formats; use the **-F** command-line parameter to force a certain format then.

COMMAND-LINE PARAMETERS

If a command-line parameter starts with a slash(/) or minus sign(-), it turns an option on; if a command-line parameter starts with a plus sign(+), it turns a specific option off. Numeric arguments to parameters can be either written in decimal or hexadecimal notation. For hexadecimal notation, prefix the number with a dollar($) sign. In the following list, all options will be shown in the form that is needed to change the default behaviour, which might be a plus or minus sign, depening on wether the option is on or off by default.

p2hex accepts the following command-line parameters:

-a

Tell P2HEX to use relative addressing in the output HEX file. By default, addresses the HEX file will be exactly the same absolute addresses as they were in the code file. This may create problems with some EPROM-burners if your code does not start at address 0. In relative mode, the address filter's start address (see the **-r** command line switch) is subtracted from all addresses specifications in the HEX file (with the exception of the entry address). Therefore, addresses in the HEX file again start at 0.

-d < <start address>-<stop address> >

Tell P2HEX that items in the address range given by the argument should be regarded as data rather than code. This option only has a meaning for the TI-DSK format that can differenciate between code and data. This option is principally obsolete since P2HEX can now directly process data from the DATA segment. It should not be used in new projects, since it may be removed in the near future.

-e <address>

Set an entry address or modify an existing one. Some HEX file formats can carry a special record for an entry address that tells a program loader where to jump after a program has been loaded. Normally, this address is generated by AS if the program's END statement has a label as argument, but this options allows to change the entry point or add one if it was forgotten in the program itself.

-F <Default|Moto|Intel|Intel16|Intel32|MOS|Tek|DSK|C>

Force a certain format for the output HEX file. By default, P2HEX will choose one depending on the target processor. Using **Default** as argument will also set this behaviour, which might be useful to revert to P2HEX's default behaviour if the default has been changed (see the discussion of the **P2HEXCMD** variable below for presetting parameters).

Moto stands for the Motorola S-Record format, which allows addresses from 16 to 32 bits in length and entry addresses. It bears its name due to the fact that every record starts with the letter S.

Intel is the "standard" Intellec-MCS8-Format for a variety of Intel 8-Bit-CPUs that almost became a standard, but was originally limited to 16-bit addresses. In later editions, the addressing capability was extended to 20 (**Intel16**) and 32 bits (**Intel32**).

MOS is a simple format introduced by the manufacturer MOS for their line of 65xx CPUs. It is limited to 16-bit addresses, just like the **Tek** format that was defined by Tektronix.

DSK is the format Texas Instruments uses for their line of 16-bit fixed-point signal processors. In contrast to all other formats, it is word-oriented and can distinguish between data and code segments. **Atmel** is the simple Hex format defined by Atmel for the AVR RISC family.

-f <number>[,<further numbers>]

Add <number> to the list of record header IDs that allow a record from a source file to be written to the target file. A certain header ID marks code for a certain target processor family; thus, this filter allows to distill code for a certain processor out of a source file that contains code for different processor families. Negation of this parameter removes certain header IDs from P2HEX's list. See the user manual of AS for a list of all possible header ID values. If P2HEX's list of header IDs is empty, no filtering will take place, i.e. all records from a source file will make it into the target file.

-i <0|1|2>

Change the terminating line for an Intel-HEX file. Normally, an Intel-HEX file is terminated with the line :**00000001FF** , but there are sources that also mention :**00000001** or :**0000000000** as the last line. The numeric argument of this parameter selects one of these opportunities, with the first one being the default. This parameter only has an effect if the target file's format is one of the Intel-HEX variants.

-l <length>

Set the maximum number of bytes per record, and therefore the maximum length of a line in the target file. Possible values are between 2 and 254 bytes, with 16 being the default.

-m <0..3>

Set one of the four Intel-Hex variants defined by Microchip for the PIC family of microcontrollers. The Default is the INHX8(0) format, which contains all words in a Lobyte-Hibyte-ordering. INHX16M(1) does just the opposite, whereas INHX8L(2) and INHX8H(3) only store the lower resp. higher bytes of each word. This parameter only has an effect if the target file's format is one of the Intel-HEX variants.

-r < <start>-<stop> >

Set a certain address range to be filtered out of the input file(s). Code that lies outside this range does not appear in the output file. As a special option, **<start>** and **<stop>** may consist of just a single dollar sign (escape this in UNIX shells!) or 0x to signify the lowest resp. highest address that occurs in the input file(s). This is also the default. Using this option will implicitly enable a second pass over all input files to find the minimum and maximum values before conversion starts, reducing the speed of P2HEX slightly.

-R <offset>

Relocate all addresses from the source file by adding the given offset.

-s

In Motorola S−Record format, force P2HEX to write a termination record after each group of data records, which may be necessary in some cases (but creates problems most of the time...)

+5

Disable output of Motorola S5−records, which contain the number of data records that were sent and therefore allow an additional level of checking. However, they are not understood by all programs and therefore might be a source of trouble.

−M <1|2|3>

Force P2HEX to use a minimum length for the address fields of Motorola S−records. For example, a value of 2 will effectively disable S1 records, and a value of 3 will force usage of S3 records. The default is 1, which enables full automatic setting of the S record length.

−avrlen <2|3>

Set the address field length of Atmel AVR Hex files to either two or three bytes (the latter is the default).

−k

Instruct P2HEX to erase the program source files after conversion.

−avrlen <2|3>

Set the width of the address field used for the Atmel hex file format. By default, 3−byte addresses will be used.

−segment <CODE|DATA|....>

Select the address space hex data is created from. By default, only records for the CODE segment (plus DATA for TI DSK) will be considered. Use this option with different arguments if the source file contains data from other address spaces. This way, multiple HEX files (one per address space) can be produced.

−cformat <string>
If C−style output is selected, this option allows to influence its structure. See the manual for details on this format string.

−q or −quiet

Enable quiet operation mode, suppressing copyright and purely informative messages. Only errors will be displayed.

PRESETTING PARAMETERS

Parameters need not neccessarily be given in the command line itself. Before processing of command line parameters starts, P2HEX will look if the **P2HEXCMD** environment variable is defined. If it exists, its contents will be treated as additional command line paramters whose syntax is absolutely equal to normal command line parameters. As exception is made if the variable’s contents start with a ’@’ sign; in such a case, the string after the ’@’ sign is treated as the name of a file that contains the options. Such a file (also called a ’key file’) has the advantage that it allows the options to be written in different lines, and it does not have a size limit. Some operating systems (like MS-DOS) do have a length limit on command lines and environment variable contents, so the key file may be your only option if you have a lot of lengthy parameters for P2HEX.

RETURN CODES

p2hex may return with the following codes:

0 no errors.

1 incorrect command line parameters.

2 I/O-error.

3 An input file had an incorrect format.

EXAMPLES

To convert a file **file1.p** fully into its HEX representation on a Unix platform, use

```
p2hex -r \$-\$ file1
```

If you additionally want to force usage of the Motorola S-Record format, use

```
p2hex -r \$-\$ -F Moto file1
```

NATIONAL LANGUAGE SUPPORT

p2hex supports national languages in the same way as AS. See the manual page for asl(1) for more information about this.

TIPS

Calling P2HEX without any arguments will print a short help listing all command line parameters.

SEE ALSO

asl(1), plist(1), pbind(1), p2bin(1)

HISTORY

P2HEX originally appeared as an AS tool in 1992, written in Borland-Pascal, and was ported to C and UNIX in 1996.

BUGS

Command line interpreters of some operating systems reserve some characters for their own use, so it might be necessary to give command line parameters with certain tricks (e.g., with the help of escape characters).

P2HEX does not have so far an opportunity to filter records by target segment. Instead, records that contain data for any other segment than CODE are completely ignored.

AUTHOR(S)

Alfred Arnold (alfred@ccac.rwth-aachen.de)