

NAME

srec_cat – manipulate EPROM load files

SYNOPSIS

srec_cat [*option...*] *filename...*

srec_cat -Help

srec_cat -VERSion

DESCRIPTION

The *srec_cat* program is used to assemble the given input files into a single output file. The use of filters (see below) allows significant manipulations to be performed by this command.

Data Order

The data from the input files is not immediately written to the output, but is stored in memory until the complete EPROM image has been assembled. Data is then written to the output file in ascending address order. The original ordering of the data (in those formats capable of random record ordering) is *not* preserved.

Data Comparison

Because input record order is not preserved, textual comparison of input and output (such as the *diff*(1) or *tkdiff*(1) commands) can be misleading. Not only can lines appear in different address orders, but line lengths and line termination can differ as well. Use the *srec_cmp*(1) program to compare two EPROM load files. If a text comparison is essential, run both files through the *srec_cat*(1) program to ensure both files to be compared have identical record ordering and line lengths.

Data Conflicts

The storing of data in memory enables the detection of data conflicts, typically caused by linker sections unintentionally overlapping.

- A warning will be issued for each address which is redundantly set to the same value.
- A fatal error will be issued if any address is set with contradictory values. To avoid this error use an **–exclude –within** filter (see *srec_input*(1)) or, to make it a warning, use the **–multiple** option (see below).
- A warning will be issued for input files where the data records are not in strictly ascending address order. To suppress this warning, use the **–disable-sequence-warning** option (see below).

These features are designed to detect problems which are difficult to debug, and detects them *before* the data is written to an EPROM and run in your embedded system.

INPUT FILE SPECIFICATIONS

Input may be qualified in two ways: you may specify a data file or a data generator. format and you may specify filters to apply to them. An input file specification looks like this:

data-file [*filter ...*]

data-generator [*filter ...*]

Data Files

Input from data files is specified by file name and format name. An input file specification looks like this:

filename [*format*] [**–ignore-checksums**]

The default format is Motorola S-Record format, but *many* others are also understood.

Data Generators

It is also possible to generate data, rather than read it from a file. You may use a generator anywhere you could use a file. An input generator specification looks like this:

–GENerate *address-range* **–data-source**

Generators include random data and various forms of constant data.

Common Manual Page

See *srec_input*(1) for complete details of input specifiers. This description in a separate manual page because it is common to more than one SRecord command.

OPTIONS

The following options are understood:

@filename

The named text file is read for additional command line arguments. Arguments are separated by white space (space, tab, newline, *etc*). There is no wildcard mechanism. There is no quoting mechanism. Comments, which start with '#' and extend to the end of the line, are ignored. Blank lines are ignored.

-Output filename [format]

This option may be used to specify the output file to be used. The special file name "-" is understood to mean the standard output. Output defaults to the standard output if this option is not used.

The *format* may be specified as:

-Absolute_Object_Module_Format

An Intel Absolute Object Module Format file will be written. (See *srec_aomf*(5) for a description of this file format.)

-Ascii_Hex

An Ascii-Hex file will be written. (See *srec_ascii_hex*(5) for a description of this file format.)

-ASM [prefix][-option...]

A series of assembler DB statements will be written.

The optional *prefix* may be specified to change the names of the symbols generated. The defaults to "eprom" if not set.

Several options are available to modify the style of output:

-Dot_Style

Use "dot" style pseudo-ops instead of words. For example `.byte` instead of the DB default.

-HEXadecimal_Style

Use hexadecimal numbers in the output, rather than the default decimal numbers.

-Section_Style

By default the generated assemble of placed at the correct address using ORG pseudo-ops. Section style output emits tables of section addresses and lengths, so the data may be related at runtime.

-A430 Generate output which is compliant to the `a430.exe` compiler as it is used, *e.g.* in IAR Embedded Workbench. This is short-hand for `-section-style -hex-style`

-CL430 Generate output which is Code Composer Essentials compliant, *i.e.* the compiler of it. This is short-hand for `-section-style -hex-style -dot-style`

-Output_Word

Generate output which is in two-byte words rather than bytes. This assumes little-endian words; you will need to use the `-Byte-Swap` filter if your target is big-endian. No attempt is made to align the words onto even address boundaries; use and input filter such as

input-file `-fill 0xFF -within input-file -range-pad 2`

to pad the data to whole words first.

-Atmel_Generic

An Atmel Generic file will be written. (See *srec_atmel_generic*(5) for a description of this file format.)

-BASic A series of BASIC DATA statements will be written.

-B-Record

A Freescale MC68EZ328 Dragonball bootstrap b-record format file will be written. (See *srec_brecord(5)* for a description of this file format.)

-Binary

A raw binary file will be written. If you get unexpected results **please** see the *srec_binary(5)* manual for more information.

-C-Array [*identifier*] [*-option...*]

A C array definition will be written.

The optional *identifier* is the name of the variable to be defined, or bogus if not specified.

-INClude

This option asks for an include file to be generated as well.

-No-CONST

This options asks for the variables to not use the const keyword (they are declared constant by default, so that they are placed into the read-only segment in embedded systems).

-C_COMpressed

These options ask for an compressed c-array whose memory gaps will not be filled.

-Output_Word

This option asks for an output which is in words not in bytes. This is little endian, so you may need to

-COsmac

An RCA Cosmac Elf format file will be written. (See *srec_cosmac(5)* for a description of this file format.)

-Dec_Binary

A DEC Binary (XXDP) format file will be written. (See *srec_dec_binary(5)* for a description of this file format.)

-Elektor_Monitor52

This option says to use the EMON52 format file when writing the file. (See *srec_emon52(5)* for a description of this file format.)

-FAIrchild

This option says to use the Fairchild Fairbug format file when writing the file. (See *srec_fairchild(5)* for a description of this file format.)

-Fast_Load

This option says to use the LSI Logic Fast Load format file when writing the file. (See *srec_fastload(5)* for a description of this file format.)

-Formatted_Binary

A Formatted Binary format file will be written. (See *srec_formatted_binary(5)* for a description of this file format.)

-FORTH [*-option*]

A FORTH input file will be written. Each line of output includes a byte value, an address, and a command.

-RAM The store command is C! This is the default.

-EEPROM

The store command is EEC!

-Four_Packed_Code

This option says to use the PFC format file when writing the file. (See *srec_fpd(5)* for a description of this file format.)

-HEX_Dump

A human readable hexadecimal dump (including ASCII) will be printed.

-Intel An Intel hex format file will be written. (See *srec_intel(5)* for a description of this file format.) The default is to emit 32-bit linear addressing; if you want 16-bit extended segment addressing use the **-address-length=2** option.

-Intel_16

An Intel-16 hex format file will be written. (See *srec_intel16(5)* for a description of this file format.)

-Memory_Initialization_File [width]

(Altera) Memory Initialization File (MIF) format will be written. The *width* defaults to 8 bits. (See *srec_mif(5)* for a description of this file format.)

-MOS_Technologies

An Mos Technologies format file will be written. (See *srec_mos_tech(5)* for a description of this file format.)

-Motorola [width]

A Motorola S-Record file will be written. (See *srec_motorola(5)* for a description of this file format.) This is the default output format. By default, the smallest possible address length is emitted, this will be S19 for data in the first 64KB; if you wish to force S28 use the **-address-length=3** option; if you wish to force S37 use the **-address-length=4** option

The optional *width* argument describes the number of bytes which form each address multiple. For normal uses the default of one (1) byte is appropriate. Some systems with 16-bit or 32-bit targets mutilate the addresses in the file; this option will imitate that behavior. Unlike most other parameters, this one cannot be guessed.

-Needham_Hexadecimal

This option says to use the Needham Electronics ASCII file format to write the file. See *srec_needham(5)* for a description of this file format.

-Ohio_Scientific

This option says to use the Ohio Scientific hexadecimal format. See *srec_os65v(5)* for a description of this format.

-SIGnetics

This option says to use the Signetics hex format. See *srec_signetics(5)* for a description of this format.

-SPAsm

This option says to use the SPASM assembler output format (commonly used by PIC programmers). See *srec_spasm(5)* for a description of this format.

-SPAsm_LittleEndian

This option says to use the SPASM assembler output format (commonly used by PIC programmers). But with the data the other way around.

-STewie

A Stewie binary format file will be written. (See *srec_stewie(5)* for a description of this file format.)

-Tektronix

A Tektronix hex format file will be written. (See *srec_tektronix(5)* for a description of this file format.)

-Tektronix_Extended

A Tektronix extended hex format file will be written. (See *srec_tektronix_extended(5)* for a description of this file format.)

-Texas_Instruments_Tagged

A TI-Tagged format file will be written. (See *srec_ti_tagged(5)* for a description of this file format.)

-Texas_Instruments_Tagged_16

A Texas Instruments SDSMAC 320 format file will be written. (See *srec_ti_tagged_16(5)* for a description of this file format.)

-Texas_Instruments_Text

This option says to use the Texas Instruments TXT (MSP430) format to write the file. See *srec_ti_txt(5)* for a description of this file format.

-VHdl [bytes-per-word [name]]

A VHDL format file will be written. The *bytes-per-word* defaults to one, the *name* defaults to *eprom*. The *etc/x_defs_pack.vhd* file in the source distribution contains an example ROM definitions pack for the type-independent output. You may need to use the *-byte-swap* filter to get the byte order you want.

-VMem [memory-width]

A Verilog VMEM format file will be written. The *memory-width* may be 8, 16, 32, 64 or 128 bits; defaults to 32 if unspecified. (See *srec_vmem(5)* for a description of this file format.) You may need to use the *-byte-swap* filter to get the byte order you want.

-WILson

A wilson format file will be written. (See *srec_wilson(5)* for a description of this file format.)

-Address_Length number

This option may be used to specify the minimum number of bytes to be used in the output to represent an address (padding with leading zeros if necessary). This helps when talking to imbecilic EPROM programmer devices which do not fully implement the format specification.

-Data_Only

This option implies the **-disable=header**, **-disable=data-count**, **-disable=exec-start-address** and **-disable=footer** options.

-ENable feature-name

This option is used to enable the output of a named feature.

Header This feature controls the presence of header records, records which appear before the data itself. Headers often, but not always, include descriptive text.

Data_Count

This feature controls the presence of data record count records, which appear after the data, and state how many data records preceded them. Usually a data integrity mechanism.

Execution_Start_Address

The feature controls the presence of execution start address records, which is where the monitor will jump to and start executing code once the hex file has finished loading.

Footer This feature controls the presence of a file termination record, one that *does not* double as an execution start address record.

Not all formats have all of the above features. Not all formats are able to optionally omit any or all the above features. Feature names may be abbreviated like command line option names.

-DISable *feature-name*

This option is used to disable the output of a named feature. See the **-enable** option for a description of the available features.

-Ignore_Checksums

The **-ignore-checksums** option may be used to disable checksum validation of input files, for those formats which have checksums at all. Note that the checksum values are still read in and parsed (so it is still an error if they are missing) but their values are not checked. Used after an input file name, the option affects that file alone; used anywhere else on the command line, it applies to all following files.

-Enable_Sequence_Warnings

This option may be used to enable warnings about input files where the data records are not in strictly ascending address order. Only one warning is issued per input. This is the default. **Note:** the output of *srec_cat(1)* is always in this order.

-Disable_Sequence_Warnings

This option may be used to disable warnings about input files where the data records are not in strictly ascending address order.

-CRLF This option is short-hand for the **-line-termination=crlf** option. For use with harebrained EPROM programmer devices which assume all the world uses Evil Bill's operating system's line termination.

-Line_Termination *style-name*

This option may be used to specify line termination style for text output. The default is to use the host operating system's default line termination style (but Cygwin behaves as if it's Unix). Use this option with caution, because it will also introduce extra (i.e. wrong) CR bytes into binary formats.

Carriage_Return_Line_Feed

Use the CRLF line termination style, typical of DOS and MS Windows.

NewLine

Use the NL line termination style, typical of Unix and Linux.

Carriage_Return

Use the CR line termination style, typical of Apple Macintosh.

All other line termination style names will produce a fatal error. Style names may be abbreviated like command line option names.

-Line_Length *number*

This option may be used to limit the length of the output lines to at most *number* characters. (Not meaningful for binary file format.) Defaults to something less than 80 characters, depending on the format.

-HEAd *string*

This option may be used to set the header comment, in those formats which support it. This option implies the **-enable=header** option.

-Execution_Start_Address *number*

This option may be used to set the execution start address, in those formats which support it. The execution start address is where the monitor will jump to and start executing code once the hex file has finished loading, think of it as a "goto" address. Usually ignored by EPROM programmer devices. This option implies the **-enable=exec-start-addr** option.

Please note: the execution start address is a different concept than the first address in memory of your data. If you want to change where your data starts in memory, use the **-offset** filter.

-MULTiple

Use this option to permit a file to contain multiple (contradictory) values for some memory locations. A warning will be printed. The last value in the file will be used. The default is for this

condition to be a fatal error.

All other options will produce a diagnostic error.

All options may be abbreviated; the abbreviation is documented as the upper case letters, all lower case letters and underscores (_) are optional. You must use consecutive sequences of optional letters.

All options are case insensitive, you may type them in upper case or lower case or a combination of both, case is not important.

For example: the arguments “-help”, “-HEL” and “-h” are all interpreted to mean the **-Help** option. The argument “-hlp” will not be understood, because consecutive optional characters were not supplied.

Options and other command line arguments may be mixed arbitrarily on the command line.

The GNU long option names are understood. Since all option names for *srec_cat* are long, this means ignoring the extra leading “-”. The “--option=value” convention is also understood.

EXIT STATUS

The *srec_cat* command will exit with a status of 1 on any error. The *srec_cat* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

srec_cat version 1.47

Copyright © 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 Peter Miller

The *srec_cat* program comes with ABSOLUTELY NO WARRANTY; for details use the '*srec_cat -Version License*' command. This is free software and you are welcome to redistribute it under certain conditions; for details use the '*srec_cat -Version License*' command.

AUTHOR

Peter Miller	E-Mail:	pmiller@opensource.org.au
^/*	WWW:	http://miller.emu.id.au/pmiller/