Topics

23 Linker Error Messages

The linker issues several types of error messages:

- Syntax and command errors
- Allocation errors
- I/O errors

This appendix discusses the three types of errors; they are listed alphabetically within each category. In these listings, the symbol (...) represents the name of an object that the linker is attempting to interact with when an error occurs.

- **Syntax/Command Errors**

  These errors are caused by incorrect use of linker directives, misuse of an input expression, or invalid options. Check the syntax of all expressions, and check the input directives for accuracy. Review the various options you are using and check for conflicts.

  absolute symbol (...) being redefined: An absolute symbol cannot be redefined.

  adding name (...) to multiple output sections: The input section is mentioned twice in the SECTIONS directive.

  ALIGN illegal in this context: Alignment of a symbol can be performed only within a SECTIONS directive.

  attempt to decrement DOT: Statements such as **.-= value** are illegal. Assignments to **dot** can be used only to create holes.

  bad fill value: The fill value must be a 16–bit constant.

  binding address for (...) redefined: Only one binding value is allowed for each section.

  blocking for (...) redefined: Only one blocking value is allowed for each section.

  can’t open filename: Specified filename cannot be opened for some reason; file doesn't exist, wrong file type, etc.

  cannot specify both binding and memory area for (...): The two are mutually exclusive. If you wish the code to be placed at a specific address, use binding only.

  cannot specify a page for a section within a GROUP

  command file nesting exceeded with file (...): Command file nesting is allowed up to 16 levels.

  -e flag does not specify a legal symbol name (...): The -e option requires a valid symbol name as an operand.

  entry point symbol (...) undefined: The symbol used with the -e option is not defined.

  errors in input - (...) not built: Previous errors prevent the creation of an output file.

  fill value for (...) redefined: Only one fill value is allowed per output section. Individual holes can be filled with different values with the section definition.

  -i path too long (...): The maximum number of characters in an -i path is 256.

illegal input character: There is a control character or other unrecognized character in the command file.

illegal memory attributes for (...): The attributes must be some combination of R, W, I, and X.

illegal operator in expression: Review legal expression operators.

illegal option within SECTIONS: The -l (lowercase L) is the only option allowed within a SECTIONS directive.

invalid path specified with -i flag: The operand of the -i flag must be a valid file or pathname.

invalid value for -f flag: must be a 2–byte constant.

invalid value for -heap flag: must be a 2–byte constant.

invalid value for -stack flag: must be a 2–byte constant.

invalid value for -v flag: must be a constant.

length redefined for memory area (...): Each memory area in a MEMORY directive can have only one length.

-m flag does not specify a valid filename: You must specify a valid filename to write the output map file to.

memory area for (...) redefined: Only one named memory allocation is allowed for each output section.

memory page for (...) redefined: Only one page allocation is allowed for each section.

memory attributes redefined for (...): Only one set of memory attributes is allowed for each output section.

missing filename on -l; use -l <filename>: The -l (lowercase L) option requires the use of a filename operand.

misuse of DOT symbol in assignment instruction: The dot symbol cannot be used in assignment statements that are outside SECTIONS directives.

no input files: The linker cannot operate without at least one input COFF file.

-o flag does specify a valid file name : string

output file has no .bss section: This is a warning. This section is usually present in a COFF file. There is no real requirement for it to be present.

output file has no .data section: This is a warning. This section is usually present in a COFF file. There is no real requirement for it to be present.

output file has no .text section: This is a warning. This section is usually present in a COFF file. There is no real requirement for it to be present.

origin missing for memory area (...)

origin redefined for memory area (...)

-r incompatible with -s (-s ignored): Since the -s option strips the relocation information and -r requests a relocatable object file, these options are in conflict with each other.

section (...) not built: The most likely cause of this is a syntax error in the SECTIONS directive.

semicolon required after assignment: There is a syntax error in the command file.

statement ignored: Caused by a syntax error in an expression.

symbol referencing errors — (...) not built

symbol (...) from file (...) being redefined: A defined symbol cannot be redefined in an assignment statement.

too many arguments - use a command file: You are limited to ten arguments on a command line, or in response to prompts.

too many -i options, 7 allowed: Additional search directories can be specified with a C_DIR or A_DIR environment variable.

type flags for (...) redefined: Only one section type is allowed per section. Note that type COPY has all of the attributes of type DSECT, so DSECT need not be specified separately.

type flags not allowed for GROUP or UNION: Special section types apply to individual sections only.

-u does not specify a legal symbol name: The -u option must specify a legal symbol name that exists in one of the files that you are linking.

unexpected EOF(end of file): Syntax error in the linker command file.

undefined symbol in expression: An assignment statement contains an undefined symbol.

unrecognized option (...): Check the list of valid options.

zero or missing length for memory area (...): Each memory range defined with the MEMORY directive must have a nonzero length.

- **Allocation Errors**

  These error messages appear during the allocation phase of linking. They generally appear if a section or group does not fit at a certain address or if the MEMORY and SECTIONS directives conflict in some way. If you are using a linker command file, check that MEMORY and SECTIONS directives allow enough room to ensure that no sections overlap and that no sections are being placed in unconfigured memory.

  alignment for (...) must be a power of 2: Section alignment must be a power of 2.

  alignment for (...) redefined: Only one alignment is allowed for each section.

  binding address (...) for section (...) is outside all memory on page (...): Each section must fall within memory configured with the MEMORY directive.

  binding address (...) for section (...) overlays (...) at (...): Two sections overlap and cannot be allocated.

  binding address (...) incompatible with alignment for section (...): The section has an alignment requirement from a .align directive or previous link. The binding address violates this requirement.

  blocking for (...) must be a power of 2: Section blocking must be a power of 2.

  can't align a section within GROUP - (...) not aligned: The entire GROUP is treated as one unit, so the GROUP can be aligned or bound to an address, but the sections making up the GROUP cannot be handled individually.

  can't align within UNION - section (...) not aligned: The entire UNION is treated as one unit, so the UNION can be aligned or bound to an address, but the sections making up the UNION cannot be handled individually.

  can't allocate (...), size ... (page ...): A section can't be allocated, because no configured memory area exists that is large enough to hold it.

  load address for uninitialized section (...) ignored: Uninitialized sections have no load addresses—only run addresses.

  load address for UNION ignored: UNION refers only to the section's run address.

  load allocation required for uninitialized UNION member (...): UNIONs refer to runtime allocation only. You must specify the load address for all sections within a UNION separately.

  no allocation allowed for uninitialized UNION member: An uninitialized section with a UNION gets its run allocation from the UNION and has no load address, so no allocation is valid for the member.

  no allocation allowed with a GROUP-allocation for section (...) ignored: The entire group is treated as one unit, so the group can be aligned or bound to an address, but the sections making up the group cannot be handled individually.

  no load address specified for (...); using run address: If an initialized section has a run address, only the section is allocated to run and load at the same address.

  no run allocation allowed for union member (...): A UNION defines the run address for all of its members; therefore, individual run allocations are illegal.

  output file (...) not executable: The output file created may have unresolved symbols or other problems stemming from other errors. This condition is not fatal.

PC–relative displacement overflow at address (...) in file (...): relocation of a PC–relative jump resulted in a jump displacement too large to encode in the instruction.

section (...) at (...) overlays at address (...): The two sections overlap and cannot be allocated.

section (...) enters unconfigured memory at address (...): A section can't be allocated because no configured memory area exists that is large enough to hold it.

section (...) not found: An input section specified in a SECTIONS directive was not found in the input file.

section (...) won't fit into configured memory: A section can't be allocated, because no configured memory area exists that is large enough to hold it.

undefined symbol (...) first referenced in file (...): Unless the -r option is used, the linker requires that all referenced symbols be defined. This condition prevents the creation of an executable output file.

- **I/O and Internal Overflow Errors:**

  The following error messages indicate that the input file is corrupt, nonexistent, or unreadable, or that the output file cannot be opened or written to. Messages in this category may also indicate that the linker is out of memory or table space. Make sure that the input file is in the correct directory and that the file system is not out of space. If the input file is corrupt, try reassembling it.

  cannot complete output file (...), write error: Usually means that the file system is out of space.

  cannot create output file (...): Usually indicates an illegal filename.

  can't find input file *filename*:

  can't open (...): The specified file does not exist.

  can't read (...)

  can't seek (...)

  can't write (...)

  can't create map file (...): Usually indicates an illegal filename.

  fail to copy (...)

  fail to read (...)

  fail to seek (...)

  fail to skip (...)

  fail to write (...)

  file (...) has no relocation information: You have attempted to relink a file that was not linked with -r.

  file (...) is of unknown type, magic number = (...): The binary input file is not a COFF file.

  illegal relocation type (...) found in section(s) of file (...): The binary file is corrupt.

  internal error (...): Indicates an internal error is in the linker.

  invalid archive size for file (...): The archive file is corrupt.

  I/O error on output file (...)

  library (...) member (...) has no relocation information

  line number entry found for absolute symbol: The input file is corrupt.

  making aux entry *filename* for symbol *n* out of sequence: The input file is corrupt.

  no string table in file *filename*: The input file is corrupt.

  no symbol map produced - not enough memory: This is a nonfatal condition that prevents the generation of the symbol list in the map file.

  overwriting aux entry filename of symbol n: The input file is corrupt.

  out of memory, aborting:

  relocation entries out of order in section (...) of file (...): The input file is corrupt.

relocation symbol not found: index (...), section (...), file (...): The input file is corrupt.

seek to (...) failed

too few symbol names in string table for archive n: The archive file is corrupt.