
APPENDIX A

COMPILER ERROR MESSAGES

This appendix lists the error messages that the compiler may generate in response to errors in your program, or in response to problems in your host system environment, such as inadequate space for temporary intermediate files that the compiler creates.

The first two passes of the compiler generally produce all user diagnostics. The initial (preprocessor) pass deals with # control lines and lexical analysis, and the next (parser) pass with everything else having to do with semantics. Only machine-dependent extensions are diagnosed in the code generator pass. If a pass produces diagnostics, later passes should not be run.

Any compiler message containing an exclamation mark ! or the word **panic** indicates that the compiler has detected an inconsistent internal state. Such occurrences are uncommon and should be reported to the maintainers.

Preprocessor (cpp80) Error Messages

-d <ident> redefined - <ident> is defined twice on command line (-d only)

bad \$if(test) - argument to \$if is not a compile time expression

bad defined syntax - the **defined** operator must be followed by an identifier, or by an identifier enclosed in parenthesis

bad #define - #define is not followed by an identifier

bad #define arguments - an ill-defined macro has been specified and later invoked

bad #ifdef - a #ifdef is not followed by an identifier

Compiler Error Messages

bad #line - #line is not followed by = or a number

bad #line syntax - #line directive is ill-formed

bad #undef - #undef is not followed by an identifier

bad #xxx - an unknown #directive has been specified

bad macro arguments - the parameters in a macro invocation are not a comma-separated list of token sequences with balanced parentheses

can't #include <file> - <file> cannot be found on the current search path for #include

can't open <file>: - input <file> cannot be opened for reading

can't open map file: <file> - <file> cannot be opened for reading (+map only)

can't read map file: <file> - cannot read 256 characters from <file> (+map only)

folded line - a logical line of more than 512 characters must be taken as two or more lines

\$if expects arguments - \$if(<expression>) has been specified with a null expression

illegal #if expression - the expression part of a #if is not a compile time expression

illegal #if syntax - expression is ill-formed

illegal ? : in #if - an ill-formed ternary operator (? :) has been specified in a #if expression

illegal character: <x> - <x> is not part of a legal C token (-x only)

illegal integer <n> - <n> is an invalid C integer

illegal float constant - an invalid float constant has been specified

illegal operator in #if - an invalid operator is present in a #if expression

illegal unary op in #if - an invalid unary operator has been specified in a #if expression

#line inside macro - a # directive occurs inside a macro invocation

Compiler Error Messages

- misplaced <command>** - a `#else`, `#elif`, or `#endif` <command> occurs without a previous `#if`, `#ifdef`, or `#ifndef`
- missing #endif** - `#if`, `#ifdef`, or `#ifndef` has no balancing `#endif` at end of input
- missing) in #if** - a (does not have a balancing) in a `#if` expression
- redefined <ident>** - the identifier <ident> has been previously defined in a `#define`
- too few macro arguments** - a macro invocation has fewer arguments than its corresponding macro declaration (+std flag only)
- too many macro arguments** - a macro invocation has more arguments than its corresponding macro declaration (+std flag only)
- unbalanced <quote>** - a " or ' <quote> is unbalanced at end of line (-x only)
- unbalanced comment in file** - the source file, or `#include` file, ends within a comment
-

Parser (cp180) Error Messages

- alias defined** - an alias must be declared before any other use or definition
- arithmetic type required** - type must be integer or floating
- bad #pragma debug** - syntax for the `#pragma` debug directive is incorrect
- bad #pragma space** - syntax for the `#pragma` space directive is incorrect
- bad (declaration)** - parenthesized declaration is in error
- bad *declaration** - pointer declaration not followed by proper <type-specifier>
- bad alias** - alias definition has bad syntax
- bad case range** - the maximum number of case labels should not exceed 512

Compiler Error Messages

bad external syntax - external declaration required, but none is recognized

bad field width - field width is negative or larger than word size

bad function argument declaration - function parameters have not been specified as identifiers separated by commas

bad function proto or argument declaration - function prototype or declaration has not been terminated by a right parenthesis

bad input file - cannot open input file

bad output file - cannot open output file

bad proto argument type - function prototype argument declaration has invalid type

cannot initialize - an initialization has been attempted for an object of unknown size

can't write - cpl80 cannot write output file

constant required - a constant integer expression is needed

const modified - an expression that could modify a const object is specified

duplicate case value - within a switch statement a case label is specified more than once

excess repeated initializers - repeated initialization in the form [<num>] <scalar> exceeds the size of the object being initialized

external name conflict - names collide when shortened or compressed to one case

function redefined - invalid function redefinition

function required - function declaration must be followed by function body

illegal % - % (modulo) operator is specified with the right operand as zero

illegal & - the "address of" operator has been applied to register, bitfield, or an rvalue expression

Compiler Error Messages

illegal / - the / (div) operator is specified with right operand as zero

illegal += - the += operator has been applied to operands with incorrect type combination

illegal actual argument - the number of arguments do not match a previous prototype declaration

illegal argument type - the actual argument type does not match the corresponding type in the prototype

illegal assignment - the operands of the assignment operator must be of equal or coercible type

illegal bitfield - an illegal type has been used in a bitfield. Legal types are either

int, unsigned int, or
char, unsigned char,

depending on the options specified.

illegal bitfield type +bf - a char bitfield has been encountered when the option has not been selected. (See Options and the +bf flag of cp180 in Appendix D.)

illegal break - a break may be used only in while, for, do, or switch statements

illegal case - a case label may only be specified inside a switch statement

illegal cast - a cast has been applied to an object that cannot be coerced to a specific type

illegal comparison - operands of comparison operators must be of scalar type

illegal continue - a continue statement may be used only in while, for, or do statements

illegal default - the default statement must be used only within a switch statement, and only once

illegal double initializer - an initializer must be a constant integer or floating expression

illegal field - bitfields are specified in a non-structure construct

illegal field initializer - initializer must be an integer constant expression

Compiler Error Messages

illegal indirection - operand of unary * is not a pointer type

illegal integer initializer - initializer must be a constant integer expression

illegal number of arguments - the number of actual arguments in a function declaration does not match that of the previous prototype declaration

illegal operand type - the operands of an arithmetic operator are not of the same or coercible types

illegal pointer initializer - initializer must be a constant integer or pointer expression

illegal pointer initializer type - initializer for a pointer object is not of a compatible type

illegal repeat count - repeat count in an initializer is not a constant integer expression

illegal return type - the type of the return expression is not coercible to that of the function

illegal selection - structure selection by . or -> has unknown field

illegal side effect - \$noside operator is specified on a construct that has side effects

illegal statement - statement is required, but none can be recognized

illegal storage class - storage class is not legal in this context

illegal switch expression - switch expressions must be an integer type

illegal type specification - type specification is not recognizable

illegal unsigned compare - an unsigned type is tested for negative values

illegal use of typedef - a typedef identifier is used in an expression

incomplete type - structure type is not followed by a tag or definition

integer type required - type must be some form of integer

Compiler Error Messages

- lvalue required** - expression must designate an object, and must not be const
- member conflict** - the same member name must have the same type and effect in all cases (no -m flag)
- member redefined** - an identifier is used more than once in structure member declaration
- missing <token>** - a syntactic element <token> is needed, but not present
- missing argument** - the number of arguments in the actual function call is less than that of its prototype declaration
- missing expression** - an expression is needed, but is not present
- missing goto label** - an identifier is needed after a goto
- missing label** - a label is not given a definition by the end of a function
- missing member name** - an identifier is needed after a -> or . operator
- not an argument** - a declaration has been specified for an argument not specified as a function parameter
- proto and arguments don't match** - the prototype function declaration provides different arguments than the actual function declaration
- redeclared argument** - a function argument has conflicting declarations
- redeclared external** - an external object or function has conflicting declarations
- redeclared enum element** - an enum element is already declared in the same scope
- redeclared local** - a local object or typedef is already declared in the same scope
- redeclared proto argument** - an identifier is used more than once in a prototype function declaration
- redeclared typedef** - a typedef is already declared in the same scope
- redefined** - a static function or object is initialized more than once

Compiler Error Messages

- redefined external** - an external function or object is defined more than once
- redefined label** - a label is specified more than once in a function
- redefined tag** - a tag is specified more than once in a given scope
- repeated type specification** - the same type modifier occurs more than once in a type specification
- scalar type required** - type must be integer, floating, or pointer
- string initializer too long** - a string is used to initialize an array of characters shorter than the string
- too many initializers** - initialization is completed for a given object before initializer list is exhausted
- too many spaces** - too many different named address spaces (@space) are used
- type conflict in conditional** - a ternary operator has been specified with middle and right operand of conflicting types
- undeclared** - an undeclared identifier appears in an expression
- undeclared argument** - all arguments to a function must be supplied by a declaration (+strict flag only)
- undefined local function** - a static function is never defined
- unexpected EOF** - last declaration is incomplete
- unknown #pragma** - a #pragma directive is not recognized
- unknown member** - field name not recognized for this struct/union
- unknown model** - a model other than s, p, d, or f has been specified (-model only)
- unused local symbol** - all variables inside a function must be used (+dead only)
- unused static** - all static variables must be used (+dead only)

Compiler Error Messages

useless expression - expression has no useful side effects, such as assignment or function call

Code Generator (cp280) Error Messages

bad builtin - the @builtin type modifier can be used only on functions.

can't initialize port object - the @port type modifier can be used only on functions.

intermediate file < release 3.0 - you are using a version of cp180 whose release number is less than 3.0. Use a newer version of cp180 to compile your program.

PANIC ! bad input file - cannot read input file.

PANIC ! bad output file - cannot create output file.

PANIC ! can't write - cannot write output file.

unknown space - you have specified an invalid space modifier of the form @xxx.

unknown space modifier - you have specified an invalid space modifier of the form @xxx.

redefined space - the version of cp180 you used to compile your program is incompatible with cp280.

All other **PANIC !** messages should never happen. If you obtain such a message, please report it with the corresponding source program to Whitesmiths or COSMIC.

