
CMS C 错误和警告消息

1. 简介

本章列出了 CMS C 2.0 编译器的错误、警告和建议性消息，以及每条消息的说明。这是完整的历史消息集合，涵盖了所有先前的 CMS C 编译器。

并不是此处显示的所有消息都与您使用的编译器版本有关。

大多数消息都指定了一个唯一编号，它出现在每条消息说明之前的括号中。在发出消息时编译器也会打印该编号。此处显示的消息按它们的编号进行排序。无编号的消息在结尾列出，并按字母顺序排序。

在错误消息对面的括号中会列出可能产生该消息的应用程序的名称。在某些情况下，会给出可能触发该错误的代码或选项的示例。在错误消息中，使用* 来表示编译器将替换的、特定于该特定错误的字符串。

请注意，C 代码或汇编源代码中的一个问题可能会触发多条错误消息。您应尝试按错误或警告的产生顺序来解决它们。

2. 错误和警告消息

2.1 消息 1-249

(1) too many errors (*)

正在执行的编译器应用程序遇到太多错误，将立即退出。

(2) error/warning (*) generated, but no description available

正在执行的编译器应用程序发出了一条消息（建议性/ 警告/ 错误），但消息说明文件中没有可供打印的说明。

(3) malformed error information on line *, in file *

编译器尝试加载选定语言的消息，但消息说明文件已损坏，无法正确读取。

(100) unterminated #if[n][def] block from line *

#if 或类似代码块未使用匹配的#endif 终止，例如：

```
#if INPUT          /* error flagged here */
void main(void)
{
    run();
}                  /* no #endif was found in this module */
```

(101) `#*` may not follow `#else`

在与某个`#else` 相同的条件块中使用了`#else` 或`#elif`。它们只能跟随在`#if` 之后，例如：

```
#ifdef FOO
    result = foo;
#else
    result = bar;
#elif defined(NEXT)    /* the #else above terminated the #if */
    result = next(0);
#endif
```

(102) `#*` must be in an `#if`

`#elif`、`#else` 或`#endif` 伪指令之前必须具有匹配的`#if` 行。如果显然存在对应的`#if` 行，则检查是否存在诸如额外的`#endif` 或未正确终止的注释之类的内容，例如：

```
#ifdef FOO
    result = foo;
#endif
    result = bar;
#elif defined(NEXT)    /* the #endif above terminated the #if */
    result = next(0);
#endif
```

(103) `#error: *`

这是一个编程人员产生的错误；有一条伪指令导致了一个有意的错误。它通常用于检查编译时定义等方面。删除该伪指令可去除错误，但请先确定该伪指令为什么出现在此处。

(104) preprocessor `#assert` failure

预处理器`#assert` 伪指令的参数求值为零。这是一个编程人员引起的错误。

```
#assert SIZE == 4    /* size should never be 4 */
```

(105) no `#asm` before `#endasm`

遇到了一个`#endasm` 操作符，但前面没有匹配的`#asm`，例如：

```
void cleardog(void)
{
    clrwdt
#endasm    /* in-line assembler ends here,
           only where did it begin? */
}
```

(106) nested #asm directives

嵌套#asm 伪指令是不合法的。请检查是否缺少或拼错了#endasm 伪指令，例如：

```
#asm
    MOVE    r0, #0aah
#asm      ; previous #asm must be closed before opening another
    SLEEP
#endasm
```

(107) illegal # directive""

编译器不理解#伪指令。它可能是错误拼写的预处理器#伪指令，例如：

```
#indef DEBUG /* oops -- that should be #undef DEBUG */
```

(108) #if[n][def] without an argument

预处理器伪指令#if、#ifdef 和#ifndef 必须有一个参数。#if 的参数应为一个表达式，而#ifdef 或#ifndef 的参数应为单个名称，例如：

```
#if      /* oops -- no argument to check */
    output = 10;
#else
    output = 20;
#endif
```

(109) #include syntax error

#include 的文件名参数的语法无效。#include 的参数必须为有效的文件名，包含在双引号""或尖括号<>中。其中不应包含空格，并且必须存在右引号或右括号。在行中不应存在注释之外的其他内容，例如：

```
#include stdio.h /* oops -- should be: #include <stdio.h> */
```

(110) too many file arguments; usage: cpp [input [output]]

CPP 应多使用两个文件参数调用。如果预处理器是由编译器驱动程序执行的，详情请联系 CMS 技术支持。

(111) redefining preprocessor macro ""

指定的宏被重新定义为不同于原始定义的内容。如果是要有意地重新定义宏，请先使用#undef 取消原始定义，例如：

```
#define ONE 1
/* elsewhere: */
/* Is this correct? It will overwrite the first definition.
*/ #define ONE one
```

(112) #define syntax error

宏定义具有语法错误。这可能是由于宏或形参名称不是以字母开始或缺少右括号)，例如：

```
#define FOO(a, 2b) bar(a, 2b) /* 2b is not to be! */
```

(113) unterminated string in preprocessor macro body

宏定义包含缺少右引号的字符串。

(114) illegal #undef argument

#undef 的参数必须为一个有效的名称。它必须以字母开始，例如：

```
#undef 6YYY /* this isn't a valid symbol name */
```

(115) recursive preprocessor macro definition of "" defined by ""

指定宏的定义方式使得在扩展它时导致递归扩展自身。

(116) end of file within preprocessor macro argument from line *

宏参数未进行终止。这可能意味着宏调用中省略了右括号。给出的行号代表宏参数的起始行，例如：

```
#define FUNC(a, b) func(a+b)
FUNC(5, 6; /* oops -- where is the closing bracket? */
```

(117) misplaced constant in #if

#if 表达式中的常量应仅出现在语法上正确的位置。该错误有可能是由于遗漏操作符而引起的，例如：

```
#if FOO BAR /* oops -- did you mean: #if FOO == BAR ? */
```

(118) stack overflow processing #if expression

在某个 #if 表达式中，预处理器填满了表达式求值堆栈。请简化表达式——它可能包含太多带括号的子表达式。

(119) invalid expression in #if line

这是内部编译器错误。详情请联系 CMS 技术支持。

(120) operator "" in incorrect context

#if 表达式中遇到了未正确放置的操作符（两个二元操作符未使用值分隔），例如：

```
#if FOO * % BAR == 4 /* what is "*" % ? */
#define BIG
#endif
```

(121) expression stack overflow at operator ""

#if 行中的表达式使用大小为 128 的堆栈进行求值。极复杂的表达式有可能使它发生溢出。请简化该表达式。

(122) unbalanced parenthesis at operator ""

#if 表达式的求值发现不匹配的括号。请检查表达式中的括号使用是否正确，例如：

```
#if ((A) + (B) /* oops -- a missing ), I think */
#define ADDED
#endif
```

(123) misplaced "?" or ":"; previous operator is ""

在 #if 表达式中遇到了未与相应操作符匹配的冒号操作符，例如：

```
#if XXX : YYY /* did you mean: #if COND ? XXX : YYY */
```

(124) illegal character "*" in #if

在`#if` 表达式中存在一个不应有的字符。有效字符包括字母、数字和构成可接受操作符的字符，例如：

```
#if YYY /* what are these characters doing here? */  
    int m;  
#endif
```

(125) illegal character (* decimal) in #if

在`#if` 表达式中存在一个不应有的不可打印字符。有效字符包括字母、数字和构成可接受操作符的字符，例如：

```
#if ^S YYY /* what is this control characters doing here? */  
    int m;  
#endif
```

(126) strings can't be used in #if

预处理器不允许在`#if` 表达式中使用字符串，例如：

```
/* no string operations allowed by the preprocessor */  
#if MESSAGE > "hello"  
#define DEBUG  
#endif
```

(127) bad syntax for defined() in #[el]if

预处理器表达式中的 `defined()` 伪函数要求其参数必须为单个名称。名称必须以字母开始，并应括在括号中，例如：

```
/* oops -- defined expects a name, not an expression */  
#if defined(a&b)  
    input = read();  
#endif
```

(128) illegal operator in #if

`#if` 表达式具有非法的操作符。请检查语法是否正确，例如：

```
#if FOO = 6 /* oops -- should that be: #if FOO == 5 ? */
```

(129) unexpected "\" in #if

反斜杠错误地处于`#if` 语句中，例如：

```
#if FOO == \34  
    #define BIG  
#endif
```

(130) unknown type "" in #[el]if sizeof()

预处理器 `sizeof()` 中使用了未知的类型。预处理器只能对基本类型或指向基本类型的指针进行 `sizeof()` 求值，例如：

```
#if sizeof(unt) == 2 /* should be: #if sizeof(int) == 2 */  
    i = 0xFFFF;  
#endif
```

(131) illegal type combination in #[el]if sizeof()

在 `#if` 表达式中的 `sizeof()` 的参数中发现非法的类型组合，例如：

```
/* To sign, or not to sign, that is the error. */  
#if sizeof(signed unsigned int) == 2  
    i = 0xFFFF;  
#endif
```

(132) no type specified in #[el]if sizeof()

在预处理器 `#if` 表达式中使用了 `sizeof()`，但未指定任何类型。在预处理器表达式中，`sizeof()` 的参数必须为有效的简单类型或指向简单类型的指针，例如：

```
#if sizeof() /* oops -- size of what? */  
    i = 0;  
#endif
```

(133) unknown type code (0x*) in #[el]if sizeof()

预处理器在 `sizeof()` 表达式求值中发生了内部错误。请检查是否存在格式错误的类型说明符。这是内部错误。详情请联系 CMS 技术支持。

(134) syntax error in #[el]if sizeof()

预处理器在 `#if` 表达式中的 `sizeof` 的参数中发现语法错误，例如：可能因为括号不匹配或类似错误，例如：

```
#if sizeof(int == 2) // oops - should be: #if sizeof(int) == 2  
    i = 0xFFFF;  
#endif
```

(135) unknown operator (*) in #if （预处理器）

预处理器试图对具有自己不理解的操作符的表达式进行求值。这是内部错误。详情请联系 CMS 技术支持。

(137) strange character "" after ## （预处理器）

在标记连接操作符 `##` 之后发现一个既不是字母也不是数字的字符。由于该操作符的结果必须为合法标记，所以操作数必须为仅包含字母和数字的标记，例如：

```
/* the ' character will not lead to a valid token */  
#define cc(a, b) a ## 'b
```

(138) strange character (*) after ##

在标记连接操作符##之后发现一个既不是字母也不是数字的不可打印字符。由于该操作符的结果必须为合法标记，所以操作数必须为仅包含字母和数字的标记，例如：

```
/* the ' character will not lead to a valid token */
#define cc(a, b) a ## 'b
```

(139) end of file in comment

在注释内遇到了文件结束符。请检查是否缺少注释结束标志，例如：

```
/* Here the comment begins. I'm not sure where I end, though
}
```

(140) can't open * file "": *

无法打开指定的命令文件进行读取。

(141) can't open * file "": *

无法创建输出文件。请确认在命令行上指定的文件的拼写和路径。

(144) too many nested #if blocks

#if 和#ifdef 等条件块多只能嵌套 32 级。

(146) #include filename too long

在查找包含文件时，所构造的文件名超出了内部缓冲区的长度。由于该缓冲区的长度为 4096 字节，通常不太可能发生这种情况。

(147) too many #include directories specified

多可以为预处理器指定 7 个目录来搜索包含文件。使用驱动程序指定的目录数太多。

(148) too many arguments for preprocessor macro

根据 C 标准，宏多只能具有 31 个参数。

(149) preprocessor macro work area overflow

宏扩展的总长度超出了内部表的长度。该表的长度通常为 32768 字节。因此任何宏扩展都不能扩展为总长度超出 32 KB。

(150) illegal "__" preprocessor macro ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(151) too many arguments in preprocessor macro expansion

在宏调用中提供了太多的参数。允许的大数量为 31。

(152) bad dp/nargs in openpar(): c = *

这是内部编译器错误。详情请联系 CMS 技术支持。

(153) out of space in preprocessor macro * argument expansion

宏参数超出了内部缓冲区的长度。该缓冲区的长度通常为 4096 字节。

(155) work buffer overflow concatenating ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(156) work buffer "" overflow

这是内部编译器错误。详情请联系 CMS 技术支持。

(157) can't allocate * bytes of memory

这是内部编译器错误。详情请联系 CMS 技术支持。

(158) invalid disable in preprocessor macro ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(159) too many calls to unget()

这是内部编译器错误。详情请联系 CMS 技术支持。

(161) control line "" within preprocessor macro expansion

在扩展宏时遇到了预处理器控制行（以#开始的行）。不应发生这种情况。

(162) #warning: *

该警告或者是用户定义的#warning 预处理器伪指令产生的，或者是驱动程序在读取映射文件时遇到了问题。如果是后者，请联系 CMS 技术支持了解详情。

(163) unexpected text in control line ignored

在控制行末尾出现额外字符时，将出现该警告。额外文本将被忽略，但会发出警告。好（根据标准 C）将文本括起来作为注释，例如：

```
#if defined(END)
#define NEXT
#endif END      /* END would be better in a comment here */
```

(164) #include filename "" was converted to lower case

#include 文件名必须先转换为小写形式，然后才能打开，例如：
#include <STDIO.H> /* oops -- should be: #include <stdio.h> */

(165) #include filename "" does not match actual name (check upper/lower case)

在 Windows 版本中，这表示要包含的文件实际上存在，并且拼写方式与#include 文件名相同；但是，每个字符的大小写并不完全匹配。例如，指定#include "code.c" 时，将会包含 Code.c（如果找到该文件）。在 Linux 版本中，如果未找到该文件，则会出现该警告。

(166) too few values specified with option ""

预处理器（CPP）-S 选项的值列表是不完整的。如果预处理器是由编译器驱动程序调用的，则不应发生这种情况。传递给该选项的值代表 char、short、int、long、float 和 double 类型的长度。

(167) too many values specified with -S option; "" unused

提供给-S 预处理器选项的值太多。请参见消息 166。

(168) unknown option ""

提供给引起错误的组件的选项无法识别。

(169) strange character (*) after ##

#之后存在意外的字符。

(170) symbol "" in undef was never defined

作为参数提供给#undef 的符号尚未定义。对于某些编译器，可以禁止该警告。该警告可以通过以下形式的代码来避免：

```
#ifdef SYM
    #undef SYM /* only undefine if defined */
#endif
```

(171) wrong number of preprocessor macro arguments for "" (* instead of *)

调用宏时使用的参数数量不正确，例如：

```
#define ADD(a, b) (a+b)
ADD(1, 2, 3) /* oops -- only two arguments required */
```

(172) formal parameter expected after #

字符串化操作符#（不应与用于预处理器控制行的前导#混淆）必须跟随形式宏参数，例如：

```
#define str(x) #y /* oops -- did you mean x instead of y? */
```

如果需要将某个标记字符串化，则需要定义一个特殊的宏来执行该操作，例如：

```
#define __mkstr__(x) #x
```

然后，在每次需要将标记转换为字符串时使用__mkstr__(token)。

(173) undefined symbol "" in #if, 0 used

#if 表达式中的某个符号不是已定义的预处理器宏。针对该表达式的目的，它的值被视为零。对于某些编译器，可以禁止该警告。示例：

```
#if FOO+BAR /* e.g. FOO was never #defined */
    #define GOOD
#endif
```

(174) multi-byte constant "" isn't portable

多字节常量是不可移植的，实际上，它会被编译器的后面阶段拒绝，例如：

```
#if CHAR == 'ab'
    #define MULTI
#endif
```

(175) division by zero in #if; zero result assumed

在#if 表达式内存在被零除，它已被视为产生零，例如：

```
#if foo/0 /* divide by 0: was this what you were intending? */
    int a;
#endif
```

(176) missing newline

行尾缺少换行符。每一行（包括后一行）末尾都必须具有换行符。该问题通常是由编辑器引入的。

(177) symbol "" in -U option was never defined

预处理器的-U 选项中指定的宏名并未初始定义，因此无法取消定义。

(179) nested comments

在发现嵌套注释时，将发出该警告。嵌套注释可能表明前一个注释结束标记缺失或格式不正确，例如：

```
output = 0; /* a comment that was left unterminated
flag = TRUE; /* next comment:
    hey, where did this line go? */
```

(180) unterminated comment in included file

在包含文件内开始的注释必须在包含文件内结束。

(181) non-scalar types can't be converted to other types

不能将结构、联合或数组转换为另一种类型，例如：

```
struct TEST test;
struct TEST * sp;
sp = test; /* oops -- did you mean: sp = &test; ? */
```

(182) illegal conversion between types

该表达式暗含不兼容类型之间的转换，如将结构类型转换为整型，例如：

```
struct LAYOUT layout;
int i;
layout = i; /* int cannot be converted to struct */
```

请注意，即使结构仅包含一个 int（举例来说），也不能将它赋予 int 变量，反之亦然。

(183) function or function pointer required

只有函数或函数指针可以作为函数调用的主体，例如：

```
int a, b, c, d;
a = b(c+d);    /* b is not a function –
                did you mean a = b*(c+d) ? */
```

(184) calling an interrupt function is illegal

不能从其他函数中调用使用 `interrupt` 限定的函数。它只能由硬件（或软件）中断调用。这是因为中断函数具有特殊的函数进入和退出代码，仅适用于从中断中进行调用。中断函数可以调用其他非中断函数。

(185) function does not take arguments

该函数没有参数，但此处使用了一个或多个参数来调用它，例如：

```
int get_value(void);
void main(void)
{
    int input;
    input = get_value(6); /* oops --
                           parameter should not be here */
}
```

(186) too many function arguments

此处的参数数量多于该函数可接受的数量。

```
void add(int a, int b);
add(5, 7, input);    /* call has too many arguments */
```

(187) too few function arguments

该函数需要的参数数量多于该调用中提供的参数数量，例如：

```
void add(int a, int b);
add(5);              /* this call needs more arguments */
```

(188) constant expression required

在该上下文中，需要一个可以在编译时求值为常量的表达式，例如：

```
int a;
switch(input) {
    case a: /* oops!
             cannot use variable as part of a case label */
        input++;
}
```

(189) illegal type for array dimension

数组维度必须为整型或枚举值。

```
int array[12.5]; /* oops -- twelve and a half elements, eh? */
```

(190) illegal type for index expression

索引表达式必须为整型或枚举值，例如：

```
int i, array[10];
i = array[3.5]; /* oops --
                exactly which element do you mean? */
```

(191) cast type must be scalar or void

类型强制转换（括在括号中的抽象类型声明符）必须指示标量（即，不是数组或结构）类型或 `void` 类型，例如：

```
lip = (long [])input; /* oops -- possibly: lip = (long *)input */
```

(192) undefined identifier "***

程序中使用了该符号，但它尚未定义或声明。如果认为它已定义，请检查是否存在拼写错误。

(193) not a variable identifier "***

该标识符不是一个变量；它可能是某种其他类型的对象，例如一个标号。

(194) ")" expected

此处要求出现一个右括号)。这可能表明在表达式中遗漏了该字符，或存在某种其他语法错误。错误将标记在代码开始变为无意义的行上。它可能是不完整表达式之后的语句，例如：

```
if(a == b /* the closing parenthesis is missing here */
    b = 0; /* the error is flagged here */
```

(195) expression syntax

该表达式的格式不正确，无法由编译器进行解析，例如：

```
a /= % b; /* oops -- possibly that should be: a /= b; */
```

(196) struct/union required

点号“.”之前需要一个结构或联合标识符，例如： `int a;`

```
a.b = 9; /* oops -- a is not a structure */
```

(197) struct/union member expected

结构或联合成员名称必须跟随在点号“.”或箭头（“->”）之后。

(198) undefined struct/union "***

指定的结构或联合标记是未定义的，例如：

```
struct WHAT what; /* a definition for WHAT was never seen */
```

(199) logical type required

用作 `if`、`while` 语句或布尔操作符（如 `!` 和 `&&`）操作数的表达式必须为标量整型，例如：

```
struct FORMAT format;
if(format) /* this operand must be a scalar type */
    format.a = 0;
```

(200) taking the address of a register variable is illegal

使用 `register` 声明的变量不能在存储器中分配存储单元，因此尝试通过应用`&`操作符获取它的地址是非法的，例如：

```
int *proc(register int in)
{
    int *ip = &in;
    /* oops -- in cannot have an address to take */
    return ip;
}
```

(201) taking the address of this object is illegal

用作`&`操作符操作数的表达式不是一个代表存储器存储单元的表达式（“左值”），因此无法定义其地址，例如：

```
ip = &8; /* oops -- you cannot take the address of a literal */
```

(202) only lvalues may be assigned to or modified

只有左值（即，直接表示可寻址存储单元的标识符或表达式）才能进行赋值或修改，例如：

```
int array[10];
int *ip;
char c;
array = ip; /* array is not a variable,
            it cannot be written to */
```

类型强制转换并不会产生一个左值，例如：

```
/* the contents of c cast to int
   is only an intermediate value */
(int)c = 1;
```

但是，可以使用指针编写以下代码：

```
*(int *)&c = 1
```

(203) illegal operation on bit variable

并不是对位变量的所有操作都受支持。该操作就是其中之一，例如：

```
bit b;
int *ip;
ip = &b; /* oops --
        cannot take the address of a bit object */
```

(204) void function can't return a value

void 函数不能返回值。任何 return 语句都不应跟随表达式，例如：

```
void run(void)
{
    step();
    return 1;
    /* either run should not be void, or remove the 1 */
}
```

(205) integral type required

该操作符要求只能为整型的操作数。

(206) Illegal use of void expression

void 表达式不具有任何值，因此不能在任何需要带值表达式的位置使用它，例如，用作算术操作符的操作数。

(207) simple type required for ""

需要一个简单类型（即，不是数组或结构）作为该操作符的操作数。

(208) operands of "" not same type

该操作符的操作数属于不同的指针，例如：

```
int * ip;
char * cp, * cp2;
cp = flag ? ip : cp2;
/* result of ? : will be int * or char * */
```

也许您的本意是以下形式的代码：

```
cp = flag ? (char *)ip : cp2;
```

(209) type conflict

该操作符的操作数属于不兼容的类型。

(210) bad size list

这是内部编译器错误。详情请联系 CMS 技术支持。

(211) taking sizeof bit is illegal

对 C bit 类型使用 sizeof 操作符是非法的。对某个类型使用 sizeof 操作符时，将给出存储该类型对象所需的字节数。因此对 bit 类型使用它毫无意义，是非法操作。

(212) missing number after pragma "pack"

`pragma pack` 需要一个十进制数作为参数。它用于指定结构中每个成员的对齐方式。使用它时需要谨慎，因为一些处理器会强制对齐，如果对奇编号边界进行字取操作，将无法正常工作，例如：

```
#pragma pack /* what is the alignment value */
```

也许您的本意是以下形式的代码：

```
#pragma pack 2
```

(214) missing number after pragma "interrupt_level"

`pragma interrupt_level` 需要一个指示中断优先级的参数。

(215) missing argument to pragma "switch"

`pragma switch` 需要使用 `auto`、`direct` 或 `simple` 作为参数，例如：

```
#pragma switch /* oops -- this requires a switch mode */
```

也许您的本意是以下形式的代码：

```
#pragma switch simple
```

(216) missing argument to pragma "psect"

`pragma psect` 需要 `oldname=newname` 形式的参数；其中，`oldname` 是编译器已知的现有 `psect` 名称，`newname` 是所需的新名称，例如：

```
#pragma psect /* oops -- this requires an psect to redirect */
```

也许您的本意是以下形式的代码：

```
#pragma psect text=specialtext
```

(218) missing name after pragma "inline"

`inline pragma` 要求其后续跟随一个函数的名称。只有代码生成器可以识别函数名时，才会对它进行扩展；其他函数不会被更改，例如：

```
#pragma inline /* what is the function name? */
```

也许您的本意是以下形式的代码：

```
#pragma inline memcpy
```

(219) missing name after pragma "printf_check"

`printf_check pragma` 要求其后续跟随一个函数的名称。它指定对函数进行 `printf` 样式的格式字符串检查，例如：

```
#pragma printf_check /* what function is to be checked? */
```

也许您的本意是以下形式的代码：

```
#pragma printf_check sprintf
```

<stdio.h>中包含了所有标准的 `printf` 样式函数的 `pragma` 伪指令。

(220) exponent expected

浮点常量在 `e` 或 `E` 之后必须具有至少一个数字，例如：

```
float f;
```

```
f = 1.234e; /* oops -- what is the exponent? */
```

(221) hexadecimal digit expected

0x 之后应至少跟随一个 HEX 数字 0-9 和 A-F 或 a-f, 例如:

```
a = 0xg6; /* oops -- was that meant to be a = 0xf6 ? */
```

(222) binary digit expected

0b 格式说明符之后要求跟随二进制数字, 例如:

```
i = 0bf000; /* oops -- f000 is not a base two value */
```

(223) digit out of range

该数值中的某个位超出了该数值的基数的范围; 例如, 在八进制数值中使用数字 8, 或在十进制数值中使用 HEX 数字 A-F。八进制数值通过以 0 开头的数字字符串表示, 而

HEX 数值则以“0X”或“0x”开头。例如:

```
int a = 058;
```

```
/* leading 0 implies octal which has digits 0 - 7 */
```

(224) illegal "#" directive

检测到非法的#预处理器伪指令。代码某处的某条伪指令可能拼写错误。

(225) missing character in character constant

单引号中缺少字符, 例如:

```
char c = "; /* the character value of what? */
```

(226) char const too long

括在单引号中的字符常量不能包含多个字符, 例如:

```
c = '12'; /* oops -- only one character can be specified */
```

(227) "." expected after ".."

可能出现两个连续点号的唯一情形是作为省略号的一部分, 而该符号必须具有三个点号。(省略号用于在函数原型中指示可变数量的参数。)

.. 的本意或者是表示省略号, 这将需要添加一个额外的点号; 或者是表示结构成员操作符, 这将需要删除一个点号。

(228) illegal character (*)

该字符在 C 代码中是非法的。有效字符包括字母、数字和构成可接受操作符的字符, 例如:

```
c = a; /* oops -- did you mean c = 'a'; ? */
```

(229) unknown qualifier "*" given to -A

这是内部编译器错误。详情请联系 CMS 技术支持。

(230) missing argument to -A

这是内部编译器错误。详情请联系 CMS 技术支持。

(231) unknown qualifier "" given to -l

这是内部编译器错误。详情请联系 CMS 技术支持。

(232) missing argument to -l

这是内部编译器错误。详情请联系 CMS 技术支持。

(233) bad -Q option ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(234) close error

这是内部编译器错误。详情请联系 CMS 技术支持。

(236) simple integer expression required

操作符@之后需要一个简单的整型表达式，用于将绝对地址与变量相关联，例如：

```
int address;
```

```
char LOCK @ address;
```

(237) function "" redefined

在该模块中遇到了一个函数的多个定义。函数重载是非法的，例如：

```
int twice(int a)
```

```
{
```

```
    return a*2;
```

```
}
```

```
/* only one prototype & definition of rv can exist */ long twice(long a)
```

```
{
```

```
    return a*2;
```

```
}
```

(238) illegal initialization

无法初始化 typedef 声明，因为它不会保留任何可以进行初始化的存储单元，例如：

```
/* oops -- uint is a type, not a variable */
```

```
typedef unsigned int uint = 99;
```

(239) identifier "" redefined (from line *)

在同一作用域中已定义了该标识符。不能重新定义它，例如：

```
int a; /* a filescope variable called "a" */
```

```
int a; /* attempting to define another of the same name */
```

请注意，同名但定义不同作用域的变量是合法的，但不建议这样做。

(240) too many initializers

该对象具有太多的初始值设定项。检查对象（数组或结构）定义中初始值设定项的数量，例如：

```
/* three elements, but four initializers */
```

```
int ival[3] = { 2, 4, 6, 8};
```

(241) initialization syntax

该对象的初始化语法不正确。请检查大括号和逗号的位置和数量是否正确，例如：

```
int iarray[10] = {'a', 'b', 'c'};
/* oops -- one two many {s */
```

(242) illegal type for switch expression

switch 操作必须具有为整型或枚举值的表达式，例如：

```
double d;
switch(d) { /* oops -- this must be integral */
    case '1.0':
        d = 0;
}
```

(243) inappropriate break/continue

发现某个 break 或 continue 语句未包围在相应的控制结构中。continue 只能在 while、for 或 do while 循环中使用，而 break 只能在这些循环或 switch 语句内使用，例如：

```
switch(input) {
    case 0:
        if(output == 0)
            input = 0xff;
    } /* oops! this should not be here; it closed the switch */
break; /* this should be inside the switch */
```

(244) "default" case redefined

在 switch 语句中只允许一个 default 标号。此处具有多个，例如：

```
switch(a) {
default: /* if this is the default case... */
    b = 9;
    break;
default: /* then what is this? */
    b = 10;
    break;
```

(245) "default" case not in switch

遇到了一个名为 default 的标号，但它不是包围在 switch 语句中。default 标号仅在 switch 语句主体内是合法的。

如果在该 default 标号之前具有一条 switch 语句，则可能说明 switch 代码中包含多余的右大括号。这将导致 switch 语句提前终止。请参见消息 246。

(246) case label not in switch

遇到了一个 `case` 标号，但不存在外围的 `switch` 语句。`case` 标号只能出现在 `switch` 语句主体内。

如果在该 `case` 标号之前具有一条 `switch` 语句，则可能说明 `switch` 代码中包含多余的右大括号。这将导致 `switch` 语句提前终止，例如：

```
switch(input) {
    case '0':
        count++;
        break;
    case '1':
        if(count>MAX)
            count= 0;
    }          /* oops -- this shouldn't be here */
    break;
    case '2':   /* error flagged here */
```

(247) duplicate label ""

在该函数中对于标号多次使用了相同的名称。请注意，标号的作用域是整个函数，而不只是包围标号的块，例如：

```
start:
    if(a > 256)
        goto end;
start:          /* error flagged here */
    if(a == 0)
        goto start; /* which start label do I jump to? */
```

(248) inappropriate "else"

遇到了无法与 `if` 语句关联的 `else` 关键字。这可能意味着缺少大括号或其他语法错误，例如：

```
/* here is a comment which I have forgotten to close...
if(a > b) {
    c = 0;
/* ... that will be closed here, thus removing the "if" */
else          /* my "if" has been lost */
    c = 0xff;
```

(249) probable missing "}" in previous block

编译器遇到了类似函数或其他声明的内容，但前面一个函数未使用右大括号结束。这可能意味着前一个函数的某个地方遗漏了右大括号，虽然它可能不是后一个，例如：

```
void set(char a) {  
    PORTA = a;  
                                /* the closing brace was left out here */  
void clear(void) /* error flagged here */  
{  
    PORTA = 0;  
}
```

2.2 消息 250-499

(251) array dimension redeclared

声明的数组维度与其先前声明具有不同的非零值。可以接受重新声明先前使用零维度声明的数组的大小，但不接受其他形式，例如：

```
extern int array[5];
int array[10];          /* oops -- has it 5 or 10 elements? */
```

(252) argument * conflicts with prototype

该函数定义的指定参数（参数 0 是左边的参数）与该函数先前的原型不一致，例如：

```
/* this is supposedly calc's prototype */
extern int calc(int, int);
int calc(int a, long int b) /* hmmm -- which is right? */
{
    /* error flagged here */
    return sin(b/a);
}
```

(253) argument list conflicts with prototype

函数定义中的参数列表与该函数的先前原型不同。请检查参数的数量和类型是否都相同。

```
extern int calc(int); /* this is supposedly calc's prototype */
int calc(int a, int b) /* hmmm -- which is right? */
{
    /* error flagged here */
    return a + b;
}
```

(254) undefined *: ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(255) not a member of the struct/union ""

该标识符不是此处使用的结构或联合类型的成员，例如：

```
struct {
    int a, b, c;
} data;
if(data.d) /* oops --
           there is no member d in this structure */
    return;
```

(256) too much indirection

指针声明只能具有 16 级的间接引用。

(257) only "register" storage class allowed

函数参数唯一允许的存储类别为 register，例如：

```
void process(static int input)
```

(258) duplicate qualifier

在该类型规范中出现了两次相同的限定符。这种情况可能是由于直接使用或通过使用 `typedef` 而导致。删除多余的限定符。例如：

```
typedef volatile int vint;
/* oops -- this results in two volatile qualifiers */
volatile vint very_vol;
```

(259) can't be qualified both far and near

同时将某个类型限定为 `far` 和 `near` 是非法的，例如：

```
far near int spooky; /* oops -- choose far or near, not both */
```

(260) undefined enum tag ""

该枚举标记尚未定义，例如：

```
enum WHAT what; /* a definition for WHAT was never seen */
```

(261) struct/union member "" redefined

在该结构或联合中已经使用了该名称的结构或联合成员，例如：

```
struct {
    int a;
    int b;
int a; /* oops -- a different name is required here */
} input;
```

(262) struct/union "" redefined

多次定义了某个结构或联合，例如：

```
struct {
    int a;
} ms;
struct {
    int a;
} ms; /* was this meant to be the same name as above? */
```

(263) members can't be functions

结构或联合的成员不能为函数。它可以为指向函数的指针，例如：

```
struct {
    int a;
    int get(int); /* should be a pointer: int (*get)(int); */
} object;
```

(264) bad bitfield type

位域的类型只能为 int（或 unsigned），例如：

```
struct FREG {
    char b0:1;    /* these must be part of an int, not char */
    char   :6;
    char b7:1;
} freg;
```

(265) integer constant expected

结构声明中的成员名称之后出现的冒号表示该成员是位域。冒号之后必须为一个整型常量，用以定义位域中的位数，例如：

```
struct {
    unsigned first: /* oops -- should be: unsigned first; */
unsigned second;
} my_struct;
```

如果它的本意是表示一个带有位域的结构，则以下给出了一个示例：

```
struct {
    unsigned first : 4;    /* 4 bits wide */
    unsigned second: 4;    /* another 4 bits */
} my_struct;
```

(266) storage class illegal

不能为结构或联合的成员指定存储类别。其存储类别取决于结构的存储类别，例如：

```
struct {
    /* no additional qualifiers can be present with members */
    static int first;
};
```

(267) bad storage class

代码生成器遇到了存储类别无效的变量定义，例如：

```
auto int foo; /* auto not permitted with global variables */
int power(static int a) /* parameters cannot be static */
{
    return foo * a;
}
```

(268) inconsistent storage class

声明具有冲突的存储类别。一个声明中应仅出现一个存储类别，例如：

```
extern static int where; /* so is it static or extern? */
```

(269) inconsistent type

一个声明中仅可出现一个基本类型，例如：

```
int float if; /* is it int or float? */
```

(270) variable can't have storage class "register"

只有函数参数或自动变量可以使用 `register` 限定符声明，例如：

```
register int gi; /* this cannot be qualified register */
int process(register int input) /* this is okay */
{
    return input + gi;
}
```

(271) type can't be long

只有 `int` 和 `float` 类型可以使用 `long` 限定。

```
long char lc; /* what? */
```

(272) type can't be short

只有 `int` 类型可以使用 `short` 修饰，例如：

```
short float sf; /* what? */
```

(273) type can't be both signed and unsigned

不能在同一声明中一起使用 `signed` 和 `unsigned` 类型修饰符，因为它们具有相反的含义，例如：

```
signed unsigned int confused; /* which is it? */
```

(274) type can't be unsigned

浮点型不能设为 `unsigned`，例如：

```
unsigned float uf; /* what? */
```

(275) "..." illegal in non-prototype argument list

省略号只能作为原型参数列表中的后一项出现。它不能独自出现，也不能出现在不具有类型的参数名之后；即，K&R 样式没有原型声明的函数定义。例如：

```
/* K&R-style non-prototyped function definition */
int kandr(a, b, ...)
    int a, b;
{
```

(276) type specifier required for prototyped argument

原型参数需要类型说明符。不能只有一个标识符。

(277) can't mix prototyped and non-prototyped arguments

函数声明只能具有全原型参数（即，括号内具有类型）或全 K&R 样式的参数（即，括号内只有名称，声明中的参数类型列在函数主体起始处之前），例如：

```
int plus(int a, b) /* oops -- a is prototyped, b is not */
int b; {
    return a + b;
}
```

(278) argument "" redeclared

指定参数在同一参数列表中声明了多次，例如：

```
/* cannot have two parameters called "a" */
int calc(int a, int a)
```

(279) initialization of function arguments is illegal

函数参数不能在声明中进行初始化。参数的初始化在调用函数时发生，并由调用函数为参数提供一个值，例如：

```
/* oops -- a is initialized when proc is called */
extern int proc(int a = 9);
```

(280) arrays of functions are illegal

不能定义一个函数数组。但是，可以定义指向函数的指针的数组，例如：

```
int * farray[](); /* oops -- should be: int (* farray[])(); */
```

(281) functions can't return functions

函数不能返回一个函数。它可以返回一个函数指针。返回函数指针的函数可以声明如下：`int (* (name()))()`。请注意正确绑定声明各个部分所需的许多括号。

(282) functions can't return arrays

函数只能返回标量（简单）类型或结构。它不能返回数组。

(283) dimension required

在多维数组中，只有高（即第一个）维度可以不赋值。之后的所有维度都必须以常量表达式表示，例如：

```
/* This should be, for example: int arr[][7] */
int get_element(int arr[2][ ])
{
    return array[1][6];
}
```

(284) invalid dimension

指定的数组维度无效。它必须大于 0。

```
int array[0]; // oops -- you cannot have an array of size 0.
```

(285) no identifier in declaration

该声明中缺少标识符。当编译器由于缺少右大括号之类错误而产生混淆时，也可能发生该错误，例如：

```
void interrupt(void) /* what is the name of this function? */
{
}
```

(286) declarator too complex

该声明符过于复杂，编译器无法进行处理。请检查声明，并寻找一种方法来简化它。如果编译器发现它太复杂，那么任何维护代码的人员也会如此。

(287) arrays of bits or pointers to bit are illegal

位的数组或指向位变量的指针是非法的，例如：

```
bit barray[10]; /* wrong -- no bit arrays */
bit * bp;       /* wrong -- no pointers to bit variables */
```

(288) only functions may be void

变量不能为 void 类型。只有函数可以为 void 类型，例如：

```
int a;
void b; /* this makes no sense */
```

(289) only functions may be qualified "interrupt"

除了函数，限定符 interrupt 不能应用于任何其他对象，例如：

```
/* variables cannot be qualified interrupt */
interrupt int input;
```

(290) illegal function qualifier(s)

对函数应用了在该上下文中没有意义的限定符。一些限定符只有在用于左值时才有意义（即 const 或 volatile）。这可能表明您忘记了指示函数应返回限定对象指针的星号*，例如：

```
const char ccrv(void) /* const * char ccrv(void) perhaps? */
{
    /* error flagged here */
    return ccip;
}
```

(291) K&R identifier "*" not an argument

该标识符出现在 K&R 样式的参数声明符中，但未列在函数名后的括号内，例如：int process(input)

```
int unput; /* oops -- that should be int input; */
{
}
```

(292) function parameter may not be a function

函数参数不能为一个函数。它可以为指向函数的指针，因此，或许声明中遗漏了“*”。

(293) bad size in index_type()

这是内部编译器错误。详情请联系 CMS 技术支持。

(294) can't allocate * bytes of memory

这是内部编译器错误。详情请联系 CMS 技术支持。

(295) expression too complex

该表达式导致编译器的内部堆栈发生溢出，应重新组合或拆分为两个表达式。

(296) out of memory

这可能是内部编译器错误。详情请联系 CMS 技术支持。

(297) bad argument (*) to tysize()

这是内部编译器错误。详情请联系 CMS 技术支持。

(298) end of file in #asm

在#asm 块内遇到文件结束符。这可能意味着#endasm 缺失或拼写错误，例如：

```
#asm
    MOV    r0, #55
    MOV    [r1], r0
}          /* oops -- where is the #endasm */
```

(300) unexpected end of file

在 C 模块中意外遇到文件结束符，例如：

```
void main(void)
{
    init();
    run();    /* is that it? What about the close brace */
```

(301) end of file on string file

这是内部编译器错误。详情请联系 CMS 技术支持。

(302) can't reopen "*: ***

这是内部编译器错误。详情请联系 CMS 技术支持。

(303) can't allocate * bytes of memory (line *)

在尝试排序和合并字符串时，解析器无法为所遇到的长字符串分配存储器。请尝试减少该模块中的字符串的数量或长度。

(306) can't allocate * bytes of memory for *

这是内部编译器错误。详情请联系 CMS 技术支持。

(307) too many qualifier names

这是内部编译器错误。详情请联系 CMS 技术支持。

(308) too many case labels in switch

该 switch 语句中具有太多的 case 标号。任何一个 switch 语句中允许的大 case 标号数量为 511。

(309) too many symbols

汇编器的符号表具有太多的符号。请减少程序中的符号数量。

(310) "]" expected

在数组声明或使用数组索引的表达式中要求存在一个右方括号，例如：

```
process(carray[idx]); /* oops --
                        should be: process(carray[idx]); */
```

(311) closing quote expected

指定字符串要求存在一个右引号。

(312) "\"" expected

解析器要求出现所指示的标记。

(313) function body expected

遇到使用 K&R 样式参数（即，括号内包含参数名，但未包含类型）的函数声明时，要求其后跟随函数主体，例如：

```
/* the function block must follow, not a semicolon */
```

```
int get_value(a, b);
```

(314) ";" expected （解析器）

语句中缺少分号。发现右大括号或关键字跟随在未使用分号终止的语句后面，例如：

```
while(a) {
    b = a-- /* oops -- where is the semicolon? */
}          /* error is flagged here */
```

注：在不处于右大括号或关键字之前的语句中遗漏分号时，通常会导致对于随后的代码（分析器假定它是原始语句的一部分）发出某种其他错误。

(315) "{" expected

此处要求出现一个左大括号。该错误可能是由于函数定义缺少左大括号而产生的，例如：

```
/* oops! no opening brace after the prototype */
void process(char c)
    return max(c, 10) * 2; /* error flagged here */
}
```

(316) `"}` expected

此处要求出现一个右大括号。该错误可能是由于某个已初始化数组缺少右大括号而产生的，例如：

```
char carray[4] = { 1, 2, 3, 4; /* oops -- no closing brace */
```

(317) `"(" expected`

此处要求出现一个左括号(。它必须是 `while`、`for`、`if`、`do` 或 `asm` 关键字之后的第一个标记，例如：

```
if a == b /* should be: if(a == b) */
    b = 0;
```

(318) `string` expected

`asm` 语句的操作数必须为括在括号内的字符串，例如：

```
asm(nop); /* that should be asm("nop");
```

(319) `while` expected (解析器)

在 `do` 语句结束处要求出现一个 `while` 关键字，例如：

```
do {
    func(i++);
} /* do the block while what condition is true? */
if(i > 5) /* error flagged here */
    end();
```

(320) `":" expected`

`case` 标号或 `default` 关键字之后缺少冒号。发生这种情况的原因通常是意外地键入分号而不是冒号，例如：

```
switch(input) {
    case 0; /* oops -- that should have been: case 0: */
        state = NEW;
```

(321) `label identifier` expected

`goto` 之后必须出现一个表示标号的标识符，例如：

```
if(a)
    goto 20;
/* this is not BASIC -- a valid C label must follow a goto */
```

(322) `enum tag` or `"{"` expected

关键字 `enum` 之后必须跟随属于或将定义为 `enum` 标记的标识符，或者跟随左大括号，例如：

```
enum 1, 2; /* should be, for example: enum {one=1, two }; */
```

(323) `struct/union tag` or `"{"` expected

在 `struct` 或 `union` 关键字之后必须跟随代表一个结构或联合的标识符或者左大括号，例如：

```
struct int a; /* this is not how you define a structure */
```

也许您的本意是以下形式的代码：

```
struct { int a;
} my_struct;
```

(324) too many arguments for printf-style format string

该格式字符串的参数太多。这是无害的，但可能表示格式字符串存在错误，例如：

```
/* oops -- missed a placeholder? */  
printf("%d - %d", low, high, median);
```

(325) error in printf-style format string

此处的格式字符串中存在错误。该字符串被解释为 printf() 样式的格式字符串，但它的语法不正确。如果不更正，它会在运行时导致意外的行为，例如：

```
printf("%l", ll); /* oops -- possibly: printf("%ld", ll); */
```

(326) long int argument required in printf-style format string

该格式说明符需要一个长整型参数。请检查格式说明符和相应参数的数量和顺序，例如：

```
printf("%lx", 2); // possibly you meant: printf("%lx", 2L);
```

(327) long long int argument required in printf-style format string

该格式说明符需要一个 long long 类型的参数。请检查格式说明符和相应参数的数量和顺序，例如：

```
printf("%llx", 2); // possibly you meant: printf("%llx", 2LL);
```

请注意，CMS C 并不直接支持 long long 整型。

(328) int argument required in printf-style format string

该 printf 样式的格式说明符需要一个整型参数。请检查格式说明符和相应参数的数量和顺序，例如：

```
printf("%d", 1.23); /* wrong number or wrong placeholder */
```

(329) double argument required in printf-style format string

对应于该参数的 printf 格式说明符为 %f 或类似形式，并需要一个浮点型表达式。请检查 printf 是否缺少或存在额外的格式说明符或参数。

```
printf("%f", 44); /* should be: printf("%f", 44.0); */
```

(330) pointer to * argument required in printf-style format string

该格式说明符需要一个指针参数。请检查格式说明符和相应参数的数量和顺序。

(331) too few arguments for printf-style format string

该格式字符串的参数太少。这将导致在运行时打印或转换一个垃圾值，例如：

```
printf("%d - %d", low);  
/* oops! where is the other value to print? */
```

(332) "interrupt_level" should be 0 to 7

pragma interrupt_level 的参数必须为 0 至 7。

```
#pragma interrupt_level 9 /* oops -- the level is too high */  
void interrupt_isr(void)  
{  
    /* isr code goes here */  
}
```

(333) unrecognized qualifier name after "strings"

向 `pragma strings` 传递了无法识别的限定符，例如：

```
/* oops -- should that be #pragma strings const ? */
#pragma strings cinst
```

(334) unrecognized qualifier name after "printf_check"

向 `#pragma printf_check` 传递了无法识别的限定符，例如：

```
/* oops -- should that be const not cinst? */
#pragma printf_check(printf) cinst
```

(335) unknown pragma ""**

遇到未知的 `pragma` 伪指令，例如：

```
#pragma rugsused myFunc w /* I think you meant regsused */
```

(336) string concatenation across lines

将连接位于两行上的字符串。请检查这是否是所需的结果，例如：

```
char * cp = "hi"
    "there"; /* this is okay,
                but is it what you had intended? */
```

(337) line does not have a newline on the end

文件中后一行的行尾缺少换行符（取决于操作系统的字符）。某些编辑器会创建此类文件，这会对于包含文件导致一些问题。ANSI C 标准要求所有源文件都仅包含完整的行。

(338) can't create * file ""**

应用程序尝试创建或打开指定的文件，但无法创建它。请检查是否所有文件路径名均正确。

(339) initializer in extern declaration

包含关键字 `extern` 的声明包含初始化。这将覆盖 `extern` 存储类别，因为要初始化一个对象，就需要定义它（即，为它分配存储单元），例如：

```
extern int other = 99; /* if it's extern and not allocated
                        storage, how can it be initialized? */
```

(340) string not terminated by null character

`char` 数组使用长度大于数组的字符串字面值进行初始化。因此，数组中没有足够的空间来安全地追加空终止符，例如：

```
char foo[5] = "12345"; /* the string stored in foo won't have
                        a null terminating, i.e.
                        foo = ['1', '2', '3', '4', '5'] */
```

(343) implicit return at end of non-void function

在已声明为返回一个值的函数中，存在使它能够到达函数主体结束而不返回一个值的执行路径。请插入一个返回某个值的 `return` 语句，或者如果该函数不返回值，则将它声明为 `void` 类型，例如：

```
int mydiv(double a, int b)
{
    if(b != 0)
        return a/b;    /* what about when b is 0? */
    /* warning flagged here */
}
```

(344) non-void function returns no value

某个函数声明为返回一个值，但其 `return` 语句未指定返回值，例如：

```
int get_value(void)
{
    if(flag)
        return val++;
    return;
    /* what is the return value in this instance? */
}
```

(345) unreachable code

永远不会执行该代码段，因为没有任何执行路径可以到达该代码段，例如：

```
while(1)            /* how does this loop finish? */
    process();
flag = FINISHED;    /* how do we get here? */
```

(346) declaration of `""` hides outer declaration

声明的某个对象与某个外部的声明（即，处于外部并位于当前函数或块之前的声明）同名。这是合法的，但可能导致在预期使用外部的变量时意外使用另一个变量，例如：

```
int input;           /* input has filescope */
void process(int a)
{
    int input;        /* local blockscope input */
    a = input;         /* this will use the local variable.
                        Is this right? */
}
```

(347) external declaration inside function

函数包含 **extern** 声明。这是合法的,但通常是不可取的,因为它会将函数声明的作用域限制在函数主体内。这意味着如果编译器稍后在同一文件中遇到该 **extern** 对象的另一处声明、使用或定义,它将不再具有先前的声明,因而无法检查这些声明是否一致。这会在链接时导致奇怪的程序行为或签名错误。它还会隐藏相同对象的任何先前声明,再次破坏编译器的类型检查。一般来说,请总是在所有其他函数之外声明 **extern** 变量和函数。例如:

```
int process(int a)
{
    /* this would be better outside the function */
    extern int away;
    return away + a;
}
```

(348) auto variable "" should not be qualified

auto 变量不应具有与之关联的 **near** 或 **far** 限定符。其存储类别由堆栈结构隐式定义。**auto** 变量可以使用 **static** 限定,但之后它将不再是 **auto** 变量。

(349) non-prototyped function declaration for ""

函数使用旧式 (K&R) 参数进行声明。好对于所有函数都使用原型声明,例如: `int process(input)`

```
int input;          /* warning flagged here */
{
}
```

好编写为以下形式:

```
int process(int input)
{
}
```

(350) unused * "" (from line *)

指示的对象从未在所编译的函数或模块中使用。该对象或者是多余的,或者先前要使用它的代码已在编译中被排除或对象名称存在拼写错误。请注意,符号 **rcsid** 和 **sccsid** 永远不会被报告为未使用。

(352) float parameter coerced to double

如果没有原型声明的函数具有声明为 **float** 的参数,编译器会将它转换为 **double float** 类型。这是因为默认的 **C** 类型转换约定规定,在浮点数传递给没有原型声明的函数时,它被转换为 **double** 类型。函数声明与该约定保持一致是非常重要的,例如:

```
double inc_flt(f) /* f will be converted to double */ float f;
                /* warning flagged here */
{
    return f * 2;
}
```

(353) sizeof external array "" is zero

外部数组大小的求值结果为零。这可能是由于数组在 `extern` 声明中不具有显式的维度。

(354) possible pointer truncation

`far` 限定的指针被赋予默认指针或 `near` 限定的指针，或默认指针被赋予一个 `near` 限定的指针。这可能会导致指针截断和信息丢失，具体取决于所使用的存储模型。

(355) implicit signed to unsigned conversion

有符号数字被赋予或以其他形式转换为较长的无符号类型。根据 ANSI C 的“值保留”规则，这将导致有符号值先通过符号扩展而扩展为一个具有目标类型长度的有符号数字，然后再转换为无符号值（这不会涉及到任何位模式变化）。因此，可能会发生意外的符号扩展。为确保不发生这种情况，需要先将有符号值转换为等效的无符号值，例如：

```
signed char sc;
unsigned int ui;
ui = sc;    /* if sc contains 0xff,
             ui will contain 0xffff for example */
```

将对 `char` 变量执行符号扩展，转换为更长的类型。如果不希望发生这种情况，请使用强制类型转换，例如：

```
ui = (unsigned char)sc;
```

(356) implicit conversion of float to integer

浮点值被赋予或以其他形式转换为整型。这可能导致浮点值发生截断。强制类型转换可以消除该警告。

```
double dd;
int i;
i = dd;    /* is this really what you meant? */
```

如果希望使用这种形式的表达式，则通过强制类型转换来指示这一点：

```
i = (int)dd;
```

(357) illegal conversion of integer to pointer

整型被赋予或以其他形式转换为指针类型。这通常意味着使用了错误的变量，但如果这确实是您希望执行的操作，请使用强制类型转换来指示编译器您希望进行该转换，警告将会被禁止。这也可能意味着您忘记了 `&` 地址操作符，例如：

```
int *ip;
int i;
ip = i;    /* oops -- did you mean ip = &i ? */
```

如果希望使用这种形式的表达式，则通过强制类型转换来指示这一点：

```
ip = (int *)i;
```

(358) illegal conversion of pointer to integer

指针被赋予或以其他形式转换为整型。这通常意味着使用了错误的变量。但如果这确实是您希望执行的操作，请使用强制类型转换来指示编译器您希望进行该转换，警告将会被消除。这也可能意味着您忘记了*解引用操作符，例如：

```
int * ip;
int i;
i = ip;    /* oops -- did you mean i = *ip ? */
```

如果希望使用这种形式的表达式，则通过强制类型转换来指示您的本意：

```
i = (int)ip;
```

(359) illegal conversion between pointer types

一种类型的指针（即指向特定类型的对象）被转换为不同类型的指针。这通常意味着使用了错误的变量，但如果这确实是您期望执行的操作，请使用强制类型转换来指示编译器您希望进行该转换，警告将会被禁止，例如：

```
long input;
char * cp;
cp = &input; /* is this correct? */
```

这是访问多字节变量内字节的常用方法。要指示这是程序的预定操作，请使用强制类型转换：

```
cp = (char *)&input; /* that's better */
```

如果指针指向具有相同类型的对象，但对象具有不同的限定符，则在指针之间进行转换也可能会出现该警告，例如：

```
char * cp;
/* yes, but what sort of characters? */
cp = "I am a string of characters";
```

如果字符串字面值的默认类型为 `const char *`，那么该警告是有效的。应将它编写为：

```
const char * cp;
cp = "I am a string of characters"; /* that's better */
```

从指针类型中省略限定符往往是灾难性的，几乎可以确定不是您所期望的。

(360) array index out of bounds

使用小于零或大于等于数组中元素数量的常量值作为索引访问数组。通过指针变量访问数组元素时，不会发出该警告，例如：

```
int i, * ip, input[10];
i = input[-2];          /* oops -- this element doesn't exist */ ip = &input[5];
i = ip[-2];             /* this is okay */
```

(361) function declared implicit int

在编译器遇到对于名称当前未定义的函数的函数调用时，编译器会自动将函数声明为 `int` 类型，具有未指定（K&R 样式）的参数。如果随后遇到了函数的定义，其类型和参数有可能不同于先前的隐式声明，从而导致编译器错误。解决方案是确保所有函数都已定义（或至少已声明），或至少在使用之前声明，好是带有原型参数。如果需要提前声明一个函数，应根据情况在它前面加上关键字 `extern` 或 `static`。例如：

```
/* I can prevent an error arising from calls below */
extern void set(long a, int b);
void main(void)
{
    /* at this point, a prototype for set() has already been seen */
    set(10L, 6);
}
```

(362) redundant "&" applied to array

地址操作符 `&` 被应用于一个数组。由于使用数组名总是会给出其地址，所以这是不必要的，已被忽略，例如：

```
int array[5];
int * ip;
/* array is a constant, not a variable; the & is redundant. */
ip = &array;
```

(363) redundant "&" or "*" applied to function address

地址操作符 `&` 被应用于一个函数。由于使用函数名总是会给出其地址，所以这是不必要的，已被忽略，例如：

```
extern void foo(void); void main(void)
{
    void(*bar)(void);
    /* both assignments are equivalent */    bar = &foo;
    bar = foo; /* the & is redundant */
}
```

(364) attempt to modify object qualified *

程序不能对声明为 `const` 或 `code` 的对象进行赋值或以其他任何方式修改。尝试修改此类对象的影响取决于特定的编译器。

```
const int out = 1234; /* "out" is read only */
out = 0;              /* oops --
                       writing to a read-only object */
```

(365) pointer to non-static object returned

该函数返回指向非静态（如自动）变量的指针。这很可能是一个错误，因为在函数返回时，与自动变量关联的存储单元会变为无效，例如：

```
char * get_addr(void)
{   char c;
    /* returning this is dangerous;
       the pointer could be dereferenced */
    return &c;
}
```

(366) operands of "*" not same pointer type

该操作符的操作数属于不同的指针类型。这可能意味着使用了错误的指针，但如果代码实际上是您所期望的，请使用强制类型转换来取消该错误消息。

(367) identifier is already extern; can't be static

该函数已声明为 **extern** 类型，可能是通过隐式声明。它现在被重新声明为 **static** 类型，但该重新声明是无效的。

```
void main(void)
{
    /* at this point the compiler assumes set is extern... */
    set(10L, 6);
}
/* now it finds out otherwise */
static void set(long a, int b)
{
    PORTA = a + b;
}
```

(368) array dimension on "*" ignored

函数参数中的数组维度已被忽略，因为参数在传递时实际上会被转换为指针。因此，可以传递任意大小的数组。或者从参数中删除大小，或者使用指针语法定义参数，例如：

```
/* param should be: "int array[]" or "int **" */
int get_first(int array[10])
{
    /* warning flagged here */
    return array[0];
}
```

(369) signed bitfields not supported

仅支持无符号位域。如果位域声明为 `int` 类型，编译器仍会将它视为无符号，例如：

```
struct {
    signed int sign: 1;    /* oops -- this must be unsigned */
    signed int value: 7;
};
```

(370) illegal basic type; int assumed

无法识别要转换为限定基本类型的强制类型转换的基本类型，基本类型被假定为 `int`，例如：

```
/* here ling is assumed to be int */
unsigned char bar = (unsigned ling) 'a';
```

(371) missing basic type; int assumed

该声明中未包含基本类型，所以假定为 `int`。该声明不是非法的，但好包含基本类型，以明确其意图，例如：

```
char c;
i;          /* don't let the compiler make assumptions, use : int i */
func();    /* ditto, use: extern int func(int); */
```

(372) ", " expected

此处要求出现一个逗号。这可能意味着在声明列表中遗漏了两个标识符之间的逗号。这也可能意味着前面紧邻的类型名称存在拼写错误，并已被解释为标识符，例如：

```
unsigned char a;
/* thinks: chat & b are unsigned, but where is the comma? */
unsigned chat b;
```

(373) implicit signed to unsigned conversion

在要求无符号类型的位置提供了有符号类型，它已被隐式强制转换为无符号类型，例如：

```
unsigned int foo = -1;
/* the above initialization is implicitly treated as:
   unsigned int foo = (unsigned) -1; */
```

(374) missing basic type; int assumed

在转换为限定基本类型的强制类型转换中缺少基本类型，并假定为 `int`，例如：

```
int i = (signed) 2; /* (signed) assumed to be (signed int) */
```

(375) unknown FNREC type "*"**

这是内部编译器错误。详情请联系 CMS 技术支持。

(376) bad non-zero node in call graph

链接器遇到了调用图中的顶层节点被调用图中的低层节点引用的情况。这可能意味着程序存在间接递归，这在使用编译堆栈时是不允许的。

(378) can't create * file ""

无法创建该类型的文件。是否该文件或具有该名称的文件已在使用？

(379) bad record type ""

这是内部编译器错误。确保目标文件是有效的目标文件。详情请联系 CMS 技术支持。

(380) unknown record type (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(381) record "" too long (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(382) incomplete record: type = *, length = *

该消息由 DUMP 或 XSTRIP 实用程序生成，指示目标文件不是有效的目标文件，或者它已被截断。详情请联系 CMS 技术支持。

(383) text record has length (*) too small

这是内部编译器错误。详情请联系 CMS 技术支持。

(384) assertion failed: file *, line *, expression *

这是内部编译器错误。详情请联系 CMS 技术支持。

(387) illegal or too many -G options

存在多个链接器-g 选项，或者-g 选项之后未跟随任何参数。这些参数用于指定如何计算段地址。

(388) duplicate -M option

第二次向链接器指定了映射文件名。如果使用的是编译器驱动程序，则不应发生该问题。如果是手动调用链接器，请确保命令行上仅存在该选项的一个实例。关于该选项的正确语法的信息。

(389) illegal or too many -O options

该链接器-o 标志是非法的，或遇到了另一个-o 选项。链接器的-o 选项必须紧跟一个文件名，不能有任何中间空格。

(390) missing argument to -P

传递给链接器的-p 选项太多，或者-p 选项未跟随任何参数。各个-p 选项的参数可以使用逗号进行组合和分隔。

(391) missing argument to -Q

-Q 链接器选项需要使用机器类型作为参数。

(392) missing argument to -U

-U（取消定义）选项需要一个参数。

(393) missing argument to -W

-W 选项（列表宽度）需要一个数值参数。

(394) duplicate -D or -H option（链接器）

第二次向链接器指定了符号文件名。如果使用的是编译器驱动程序，则不应发生该问题。如果是手动调用链接器，请确保命令行上仅存在这两个选项之一的一个实例。

(395) missing argument to -J

必须在-j 链接器选项之后指定中止之前的大错误数量。

(397) usage: hlink [-options] files.obj files.lib

命令行链接器的用法不正确。如果要直接调用链接器。否则，这可能是内部编译器错误，详情请联系 CMS 技术支持。

(398) output file can't be also an input file

链接器检测到尝试将其输出文件覆盖写入其输入文件之一。无法执行该操作，因为它需要同时读取和写入输入和输出文件。

(400) bad object code format

这是内部编译器错误。目标文件的目标代码格式无效。请确保它是有效的目标文件。详情请联系 CMS 技术支持。

(402) bad argument to -F

为 objtohex 的-F 选项提供了无效的参数。这可能是内部编译器错误，详情请联系 CMS 技术支持。

(403) bad -E option: ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(404) bad maximum length value to -<digits>

OBJTOHEX -n,m HEX 长度/ 舍入选项的第一个值无效。

(405) bad record size rounding value to -<digits>

OBJTOHEX -n,m HEX 长度/ 舍入选项的第二个值无效。

(406) bad argument to -A

这是内部编译器错误。详情请联系 CMS 技术支持。

(407) bad argument to -U

这是内部编译器错误。详情请联系 CMS 技术支持。

(408) bad argument to -B

该选项需要以 8、10 或 16 为基数的整型参数。这可能是内部编译器错误，详情请联系 CMS 技术支持。

(409) bad argument to -P

该选项需要以 8、10 或 16 为基数的整型参数。这可能是内部编译器错误，详情请联系 CMS 技术支持。

(410) bad combination of options

提供给 OBJTOHEX 的选项组合是无效的。

(412) text does not start at 0

某些内容中的代码必须从零开始。此处不是如此。

(413) write error on ""**

指定文件发生写入错误。这可能意味着磁盘空间已用尽。

(414) read error on ""**

链接器尝试读取该文件时遇到一个错误。

(415) text offset too low in COFF file

这是内部编译器错误。详情请联系 CMS 技术支持。

(416) bad character (*) in extended TEKHEX line

这是内部编译器错误。详情请联系 CMS 技术支持。

(417) seek error in ""**

这是内部编译器错误。详情请联系 CMS 技术支持。

(418) image too big

这是内部编译器错误。详情请联系 CMS 技术支持。

(419) object file is not absolute

传递给 OBJTOHEX 的目标文件中具有重定位项。这可能表明它是错误的目标文件，或者为链接器或 OBJTOHEX 提供了无效的选项。来自汇编器的目标输出文件是可重定位的，不是绝对的。链接器的目标文件输出是绝对的。

(420) too many relocation items

这是内部编译器错误。详情请联系 CMS 技术支持。

(421) too many segments

这是内部编译器错误。详情请联系 CMS 技术支持。

(422) no end record

该目标文件没有结束记录。这可能意味着它不是一个目标文件。如果目标文件是由编译器生成的，请联系 CMS 技术支持。

(423) illegal record type

目标文件中存在错误。这说明目标文件无效或链接器中发生了内部错误。如果目标文件是由编译器创建的，请联系 CMS 技术支持了解详情。

(424) record too long

这是内部编译器错误。详情请联系 CMS 技术支持。

(425) incomplete record

传递给 OBJTOHEX 或库管理器的目标文件已损坏。详情请联系 CMS 技术支持。

(427) syntax error in checksum list

由 OBJTOHEX 读取的校验和列表中存在语法错误。校验和列表是为了响应某个选项从标准输入中读取的。

(428) too many segment fixups

这是内部编译器错误。详情请联系 CMS 技术支持。

(429) bad segment fixups

这是内部编译器错误。详情请联系 CMS 技术支持。

(430) bad checksum specification

提供给 OBJTOHEX 的校验和列表的语法不正确。

(431) bad argument to -E

该选项需要以 8、10 或 16 为基数的整型参数。如果要直接调用 objtoexe，请检查该参数。否则，这可能是内部编译器错误，详情请联系 CMS 技术支持。

(432) usage: objtohex [-ssymfile] [object-file [exe-file]]

命令行工具 objtohex 的用法不正确。这可能是内部编译器错误，详情请联系 CMS 技术支持。

(434) too many symbols (*)

符号表中存在太多的符号，其限制为* 个符号。请将一些全局符号更改为局部符号，以减少符号数量。

(435) bad segment selector ""

链接器的段规范选项（-G）是无效的，例如：

-GA/f0+10

是否忘记了基数？

-GA/f0h+10

(436) psect "" re-orged

该 psect 多次指定了其起始地址。

(437) missing "=" in class spec

类规范需要一个=符号，例如，Ctext=ROM。

(438) bad size in -S option

-S 规范中给出的地址无效：它应为有效的数字，使用十进制、八进制或十六进制基数。基数通过后缀 O（八进制）或 H（十六进制）来指定。也可以使用前缀 0x 来指示十六进制。大小写对于任何数字或基数都不重要。十进制是默认设置，例如：

-SCODE=f000

是否忘记了基数？

-SCODE=f000h

(439) bad -D spec: "*"

-D 规范（为类提供 delta 值）的格式是无效的，例如：

-DCODE

该类的 delta 值是什么？也许您的本意是以下形式的代码：

-DCODE=2

(440) bad delta value in -D spec

为-D 规范提供的 delta 值是无效的。该值应为以 8、10 或 16 为基数的整数。

(441) bad -A spec: "*"

-A 规范（为链接器提供地址范围）的格式是无效的，例如：

-ACODE

该类的范围是什么？也许您的本意是：

-ACODE=0h-1fffh

(442) missing address in -A spec

-A 规范（为链接器提供地址范围）的格式是无效的，例如：

-ACODE=

该类的范围是什么？也许您的本意是：

-ACODE=0h-1fffh

(443) bad low address "*" in -A spec

-A 规范中给出的低地址是无效的：它应为有效的数字，使用十进制、八进制或十六进制基数。基数通过后缀 O（八进制）或 H（十六进制）来指定。也可以使用前缀 0x 来指示十六进制。大小写对于任何数字或基数都不重要。十进制是默认设置，例如：

-ACODE=1fff-3fffh 是否忘记了基数？

-ACODE=1fffh-3fffh

(444) expected "-" in -A spec

链接器选项-A 中的高地址和低地址之间需要一个减号-，例如：

-AROM=1000h

也许您的本意是：

-AROM=1000h-1fffh

(445) bad high address "*" in -A spec

-A 规范中给出的高地址是无效的：它应为有效的数字，使用十进制、八进制或十六进制基数。基数通过后缀 O（八进制）或 H（十六进制）来指定。也可以使用前缀 0x 来指示十六进制。大小写对于任何数字或基数都不重要。十进制是默认设置，例如：

-ACODE=0h-ffff 是否忘记了基数？

-ACODE=0h-ffffh

(446) bad overrun address "*" in -A spec

-A 规范中给出的溢出地址是无效的：它应为有效的数字，使用十进制、八进制或十六进制基数。基数通过后缀 O（八进制）或 H（十六进制）来指定。也可以使用前缀 0x 来指示十六进制。大小写对于任何数字或基数都不重要。十进制是默认设置，例如：

-AENTRY=0-0FFh-1FF

是否忘记了基数？

-AENTRY=0-0FFh-1FFh

(447) bad load address "*" in -A spec

-A 规范中给出的装入地址是无效的：它应为有效的数字，使用十进制、八进制或十六进制基数。基数通过后缀 O（八进制）或 H（十六进制）来指定。也可以使用前缀 0x 来指示十六进制。大小写对于任何数字或基数都不重要。十进制是默认设置，例如：

-ACODE=0h-3fffh/a000

是否忘记了基数？

-ACODE=0h-3fffh/a000h

(448) bad repeat count "*" in -A spec

-A 规范中给出的重复计数是无效的，例如：

-AENTRY=0-0FFhxf

是否忘记了基数？

-AENTRY=0-0FFhxfh

(449) syntax error in -A spec: *

-A 规范是无效的。有效的-A 规范应为以下形式：

-AROM=1000h-1FFFh

(450) psect "*" was never defined

-P 选项中列出了该 psect，但它未在程序内的任何模块中定义。

(451) bad psect origin format in -P option

-P 选项中的起点格式不是格式有效的十进制、八进制或十六进制数字，它也不是现有 psect 的名称。十六进制数字必须跟随后缀 H，例如：

-pbss=f000

是否忘记了基数？

-pbss=f000h

(452) bad "+" (minimum address) format in -P option

链接器-p 选项中的低地址规范存在格式错误, 例如:

-pbss=data+f000

是否忘记了基数?

-pbss=data+f000h

(453) missing number after "%" in -P option

-p 选项中的%操作符(用于舍入边界)必须跟随一个数字。

(454) link and load address can't both be set to "." in -P option

psect 的链接和装入地址都使用点号字符指定。这两个地址中只有一个地址可以使用这种方式指定, 例如:

-Pmypsect=1000h/.

-Pmypsect=./1000h

这两个选项都是有效的, 并且是等效的。但是, 以下用法具有歧义:

-Pmypsect=./.

该 psect 的链接或装入地址是什么?

(455) psect "*" not relocated on 0x* byte boundary

该 psect 未被重定位到所需的边界上。检查 psect 是否可重定位, 并根据需要更正-p 选项。

(456) psect "*" not loaded on 0x* boundary

该 psect 具有重定位要求, 但-p 选项给出的装入地址无法满足它。例如, 如果某个 psect 必须处于 4 KB 边界上, 则不能将它设为在 100H 处开始。

(459) remove failed, error: *, *

输出文件的创建在删除中间文件时发生失败。

(460) rename failed, error: *, *

输出文件的创建在重命名中间文件时发生失败。

(461) can't create * file ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(464) missing key in avmap file

这是内部编译器错误。详情请联系 CMS 技术支持。

(465) undefined symbol "*" in FNBREAK record

链接器在某个不可重入函数的 FNBREAK 记录中发现未定义的符号。如果它不是手写的汇编代码, 请联系 CMS 技术支持。

(466) undefined symbol "*" in FNINDIR record

链接器在某个不可重入函数的 FNINDIR 记录中发现未定义的符号。如果它不是手写的汇编代码, 请联系 CMS 技术支持。

(467) undefined symbol "" in FNADDR record

链接器在某个不可重入函数的 FNADDR 记录中发现未定义的符号。如果它不是手写的汇编代码，请联系 CMS 技术支持。

(468) undefined symbol "" in FNCALL record

链接器在某个不可重入函数的 FNCALL 记录中发现未定义的符号。如果它不是手写的汇编代码，请联系 CMS 技术支持。

(469) undefined symbol "" in FNROOT record

链接器在某个不可重入函数的 FNROOT 记录中发现未定义的符号。如果它不是手写的汇编代码，请联系 CMS 技术支持。

(470) undefined symbol "" in FNSIZE record

链接器在某个不可重入函数的 FNSIZE 记录中发现未定义的符号。如果它不是手写的汇编代码，请联系 CMS 技术支持。

(471) recursive function calls:

这些函数（或这个函数）相互递归调用。这些函数中的一个或多个具有静态分配的局部变量（编译堆栈）。请使用 `reentrant` 关键字（如果该编译器支持它）或重新进行编码以避免递归，例如：

```
int test(int a)
{
    if(a == 5) {
        /* recursion cannot be supported by some compilers */
        return test(a++);
    }
    return 0;
}
```

(472) non-reentrant function "" appears in multiple call graphs: rooted at "" and ""

主干代码和中断代码都会调用该函数。请使用 `reentrant` 关键字（如果该编译器支持它）或重新进行编码以避免使用局部变量或参数，或者复制该函数，例如：

```
void interrupt my_isr(void)
{
    scan(6);    /* scan is called from an interrupt function */
}

void process(int a)
{
    scan(a);    /* scan is also called from main-line code */
}
```

(473) function "" is not called from specified interrupt_level

永远不会从相同中断级别的中断函数中调用所指示的函数，例如：

```
#pragma interrupt_level 1
void foo(void) {
    ...
}
#pragma interrupt_level 1
void interrupt bar(void)
{
    // this function never calls foo()
}
```

(474) no psect specified for function variable/argument allocation

从未发现 FNCONF 汇编器伪指令（向链接器指定关于自动/ 参数块的信息）。它在标准运行时文件中提供（如需要）。该错误可能意味着未链接正确的运行时启动模块。如果运行时启动模块是手写的，请确保已使用 FNCONF 伪指令。

(475) conflicting FNCONF records

链接器发现两条冲突的 FNCONF 伪指令。该伪指令应仅指定一次，并包含在通常链接到每个程序中的标准运行时启动代码中。

(476) fixup overflow referencing ** (location 0x* (0x*+*), size *, value 0x*)

链接器被要求重定位（fixup）某个项目，但它在重定位之后将无法装回到空间中。更多信息，请参见下面的错误消息 (477)。

(477) fixup overflow in expression (location 0x* (0x*+*), size *, value 0x*)

fixup 是一个由链接器执行的过程，该过程会将汇编指令中对变量等的符号引用替换为一个绝对值。它是在将 psect（程序段或块）定位到目标器件上的可用存储器中之后进行的。如果对于某个符号确定的值太大，无法装入汇编指令分配的空间内，将会发生 fixup 溢出。例如，如果汇编指令使用 8 位字段来存放地址，而链接器确定用来表示该地址的符号具有值 0x110，则显然无法将该值插入指令。

该错误的原因有很多，但手写汇编器代码总是第一嫌疑对象。写得很差的 C 代码也可能会生成产生 fixup 溢出错误的汇编代码。假设具有以下错误消息。

```
main.obj: 8: Fixup overflow in expression (loc 0x7FD (0x7FC+1),      size 1, value 0x1FC)
```

这表明导致问题的文件为 main.obj。它通常为编译 main.c 或 main.as 产生的输出。它指示您应查看的文件。下一个数字（在该示例中为 8）是导致问题的目标文件中的记录编号。如果使用 DUMP 实用程序来检查目标文件，可以确定该记录；但是，通常不需要这么做。

指令所在的存储单元 loc（0x7FD）、指令中对应于值的字段的长度 size（1，以字节为单位），以及符号表示的实际值 value（0x1FC），通常是追踪错误原因唯一需要的信息。请注意，如果长度不是 8 位的倍数，将会向上舍入为接近的字节长度；例如，指令中的 7 位空间将显示为 1 字节。

生成相应模块的汇编列表文件。查找错误消息中指定的地址。

```
7    07FC  0E21  MOVLW 33
8    07FD  6FFC  MOVWF _foo
9    07FE  0012  RETURN
```

并且为了进行确认，在同一文件底部的符号表中查找该地址处的汇编指令引用的符号。

Symbol Table Fri Aug 12 13:17:37 2004

```
_foo 01FC    _main 07FF
```

在该示例中，导致问题的指令可接受存储区的 8 位偏移量，但显然地址 0x1FC 超出了这个长度。也许应将该指令编写为：

```
MOVWF (_foo&0ffh)
```

它会屏蔽掉包含存储区信息的地址高位。

如果导致该错误的汇编指令是由编译器生成的，则在汇编列表文件根据该指令查找导致错误的文件，以确定是哪条 C 语句生成了该指令。然后，您需要通过检查 C 代码来确定可能的错误。未正确限定的指针是常见的触发原因。

(478) * range check failed (location 0x* (0x*+*), value 0x* > limit 0x*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(479) circular indirect definition of symbol ""

指定符号已被设为等同于某个外部符号，后者又反过来被设为等同于第一个符号。

(480) function signatures do not match: * (*): 0x*/0x*

指定函数在不同模块中具有不同的签名。这意味着它被以不同的形式声明；例如，它可能在一个模块中声明了原型，但未在另一个中声明。在两个指定模块中，检查函数的可见声明，并确保它们是兼容的，例如：

```
extern int get_value(int in);
/* and in another module: */
/* this is different to the declaration */ int get_value(int in, char type)
{
```

(481) common symbol "" psect conflict

某个公共符号被定义为处于多个 psect 中。

(482) symbol "" is defined more than once in ""

在多个位置定义了该符号。如果某个符号在同一模块中多次定义，则汇编器会发出该错误，例如：

```
_next:
    MOVE r0, #55
    MOVE [r1], r0
_next:          ; oops -- choose a different name
```

如果符号（C 或汇编）在不同模块中多次定义，则链接器会发出该警告。错误消息中会给出模块的名称。请注意，在编译之后，C 标识符通常会在其名称前加上下划线。

(483) symbol "" can't be global

这是内部编译器错误。详情请联系 CMS 技术支持。

(484) psect "" can't be in classes "" and ""

一个 psect 不能处于多个类中。这或者是由于汇编模块对于 PSECT 伪指令使用了冲突的 class=选项，或者是对于链接器使用了冲突的-C 选项而导致的，例如：

```
psect final,class=CODE
finish:
/* elsewhere: */
psect final,class=ENTRY
```

(485) unknown "with" psect referenced by psect ""

指定 psect 使用 psect with 标志与某个 psect 一起放置。用于放置它的 psect 不存在，例如：

```
psect starttext,class=CODE,with=rextext
; was that meant to be with text?
```

(486) psect "" selector value redefined

该 psect 的选择器值定义了多次。

(487) psect "" type redefined: /*

不同模块为该 psect 定义了不同的类型。这可能意味着您试图链接不兼容的目标模块；例如，将 386 平面模型代码与 8086 实模式代码进行链接。

(488) psect "" memory space redefined: /*

某个全局 psect 被定义为处于两个不同的存储空间。请重命名其中一个 psect，或者如果它们是同一 psect，则使用 space psect 标志将它们放在同一存储空间中，例如：

```
psect spdata,class=RAM,space=0
ds 6
; elsewhere:
psect spdata,class=RAM,space=1
```

(489) psect "" memory delta redefined: /*

某个全局 psect 被定义了两个不同的 delta 值，例如：

```
psect final,class=CODE,delta=2
finish:
; elsewhere:
psect final,class=CODE,delta=1
```

(490) class "" memory space redefined: /*

某个类被定义为处于两个不同的存储空间。请重命名其中一个类，或者如果它们是同一个类，则将它们放在同一存储空间中。

(491) can't find 0x* words for psect "*" in segment "**"

链接器执行的主要任务之一是将它从程序生成的代码和数据块（或 psect）定位到目标器件的可用存储器中。该错误指示链接器无法找到足以容纳其中一个 psect 的可用存储区。该错误消息会指示链接器尝试定位的 psect 的名称和段名，后者通常是使用链接器-A 选项定义的类的名称。通常，名称为 text 或包含 text 的 psect 与程序代码相关。bss 或 data 之类的名称指代变量块。该错误的可能原因有两个。

- ◆ 第一个原因是，程序或程序数据的长度超出了选定器件上的空间总量。即，器件存储器的某些部分已完全填满。如果是这种情况，则必须减小所指定 psect 的长度。
- ◆ 该消息的第二个原因是要定位的 psect 所需的存储器总量是足够的，但该存储器被碎片化，大的连续块太小而无法容纳该 psect。链接器无法在这种情况下拆分 psect。即，链接器无法将 psect 的一部分放在一个位置，将其他部分放在别的位置。因此，链接器必须能够为每个 psect 寻找足够大的连续存储器块。如果这是错误的原因，则必须将 psect 拆分为较小的 psect（如可能）。

要确定哪些存储器仍然可用，可以生成映射文件并在其中进行查找：搜索字符串 **UNUSED ADDRESS RANGES**。在该标题下，查找错误消息中指定的段的名称。如果该名称不存在，则说明可用于该 psect 的所有存储器都已分配。如果它存在，则在该段下对于每个可用存储器块都会指定一个地址范围。确定每个块的大小，并将它与错误消息中指定的字数进行比较。

通过使用所有编译器优化或重新构造程序，可以减小包含代码的 psect。如果必须将某个代码 psect 拆分为两个或更多的小 psect，就会需要将一个函数拆分为两个或多个更小的函数（它们可以相互调用）。这些函数可能需要放在新的模块中。

在调用编译器优化时，包含数据的 psect 可能会减小，但效果不会那么显著。程序可能需要重新编写，从而使它需要的变量减少。如果必须更改默认的链接器选项，可以使用驱动程序-L- 选项间接通过驱动程序实现。如果数据 psect 需要拆分成更小的 psect，变量的定义将需要移动到新的模块中，或更均地分布在现有模块中。自动变量的存储器分配完全由编译器进行处理。除了减少使用的这些变量的数量之外，编程人员几乎无法控制它们的操作。

无论编译的代码使用的是硬件堆栈还是编译堆栈，都适用这一点。

例如，在接收到以下消息之后：

Can't find 0x34 words (0x34 withtotal) for psect text in segment CODE (error)

在映射文件中查找未用存储器的范围。

UNUSED ADDRESS RANGES

CODE	00000244-0000025F
00001000-0000102f	
RAM	00300014-00301FFB

在 CODE 段中，有一个块具有 0x1c (0x25f-0x244+1) 字节的可用空间，另一个块具有 0x30 的可用空间。它们都不够大，无法容纳 0x34 字节长的 psect text。但请注意，可用的存储器总量大于 0x34 字节。

(492) attempt to position absolute psect "*" is illegal

该 psect 是绝对的，不应在 -P 选项中指定它的地址。删除 abs psect 标志或删除 -P 链接器选项。

(493) origin of psect "*" is defined more than once

该 psect 的起点定义了多次。很可能存在多个指定了该 psect 的 -p 链接器选项。

(494) bad -P format "*/"

传递给链接器的-P 选项的格式不正确。该选项用于指定 psect 的位置，例如：

-Ptext=10g0h

也许您的本意是：

-Ptext=10f0h

(495) use of both "with=" and "INCLASS/INCLASS" allocation is illegal

如果在定义某个 psect 时，还使用了 with psect 标志，则在类中指定该 psect 的链接和位置是不合法的。

(497) psect "*" exceeds max size: *h > *h

psect 的字节数多于使用 size psect 标志指定的大允许值。

(498) psect "*" exceeds address limit: *h > *h

psect 的高地址超出了使用 limit psect 标志对它设置的限制。该 psect 需要链接到不同位置或 psect 中存在太多的代码/ 数据。

(499) undefined symbol:

随后的符号在链接时是未定义的。这可能是由于拼写错误或未能链接相应模块而导致的。

2.3 消息 500-749

(500) undefined symbols:

随后的符号列表在链接时是未定义的。这些错误可能是由于拼写错误或未能链接相应模块而导致的。

(501) program entry point is defined more than once

传递给链接器的目标文件中定义了多个入口点。结束入口点在 **END** 伪指令之后指定。运行时启动代码定义入口点，例如：

```
powerup:
    goto start
    END powerup ; end of file and define entry point
; other files that use END should not define another entry point
```

(502) incomplete * record body: length = *

一个目标文件包含具有非法长度的记录。这可能意味着该文件已被截断或不是一个目标文件。详情请联系 CMS 技术支持。

(503) ident records do not match

传递给链接器的目标文件不具有匹配的 **ident** 记录。这意味着它们针对的是不同的器件类型。

(504) object code version is greater than *.*

目标模块的目标代码版本高于已知链接器可处理的高版本。请检查是否使用了正确的链接器。如果尚未对链接器应用补丁，请联系 CMS 技术支持。

(505) no end record found in object file

目标文件中未包含结束记录。这可能意味着该文件已损坏或不是一个目标文件。如果目标文件是由编译器生成的，请联系 CMS 技术支持。

(506) object file record too long: *.*

这是内部编译器错误。详情请联系 CMS 技术支持。

(507) unexpected end of file in object file

这是内部编译器错误。详情请联系 CMS 技术支持。

(508) relocation offset (*) out of range 0..*-1

这是内部编译器错误。详情请联系 CMS 技术支持。

(509) illegal relocation size: *

链接器读取的目标代码格式中存在错误。这或者意味着您使用的链接器已过时，或者汇编器或链接器中发生了内部错误。如果目标文件是由编译器创建的，请联系 CMS 技术支持了解详情。

(510) complex relocation not supported for -R or -L options

向链接器传递了 -R 或 -L 选项，但文件包含复杂的重定位。

(511) bad complex range check

这是内部编译器错误。详情请联系 CMS 技术支持。

(512) unknown complex operator 0x*

这是内部编译器错误。详情请联系 CMS 技术支持。

(513) bad complex relocation

链接器被要求执行语法不正确的复杂重定位。可能意味着目标文件已损坏。

(514) illegal relocation type: *

一个目标文件包含具有非法重定位类型的重定位记录。这可能意味着该文件已损坏或不是一个目标文件。如果目标文件是由编译器创建的，请联系 CMS 技术支持了解详情。

(515) unknown symbol type *

这是内部编译器错误。详情请联系 CMS 技术支持。

(516) text record has bad length: *-*(-*+1) < 0

这是内部编译器错误。详情请联系 CMS 技术支持。

(520) function "*" is never called

该函数永远不会被调用。这可能并不表示存在问题，但可以通过删除它来节省空间。如果您认为应调用该函数，请检查您的源代码。一些汇编库程序永远不会被调用，虽然它们实际上会执行。在这种情况下，这些程序将按特殊顺序进行链接，从而使程序执行从一个程序直落到下一个。

(521) call depth exceeded by function ""**

调用图显示函数的嵌套深度大于指定的深度。

(522) library "*" is badly ordered

该库中的顺序很差。它仍然可以正确链接，但如果更好地安排其中的顺序，链接速度会更快。

(523) argument to -W option (*) illegal and ignored

链接器选项-w 的参数超出范围。该选项控制两个功能。对于警告级别，其范围为-9 至 9。对于映射文件宽度，其范围为大于等于 10。

(524) unable to open list file "*": *

无法打开指定的列表文件。链接器将尝试 fixup 列表文件，从而使它包含绝对地址。请确保在编译阶段生成汇编列表文件。或者，从链接步骤中删除汇编列表文件生成选项。

(525) too many address (memory) spaces; space (*) ignored

地址空间数量的限制（使用 PSECT 汇编器伪指令指定）当前为 16。

(526) psect "*" not specified in -P option (first appears in "")**

该 psect 未在链接器的-P 或-A 选项中指定。它已被链接到程序末尾，这可能不是您所希望的位置。

(528) no start record; entry point defaults to zero

传递给链接器的所有目标文件均不包含起始记录。程序的起始地址已被设置为零。这可能是无害的，但建议在启动模块中通过使用 END 伪指令定义起始地址。

(529) usage: objtohex [-Ssymfile] [object-file [HEX-file]]

命令行工具 objtohex 的用法不正确。这可能是内部编译器错误，详情请联系 CMS 技术支持。

(593) can't find 0x* words (0x* withtotal) for psect "*" in segment ""**

请参见消息 (491)。

(594) undefined symbol:

随后的符号在链接时是未定义的。这可能是由于拼写错误或未能链接正确的模块而导致的。

(595) undefined symbols:

随后的符号列表在链接时是未定义的。这些错误可能是由于拼写错误或未能链接正确的模块而导致的。

(596) segment "*" (*-*) overlaps segment "*" (*-*)

指定的段具有重叠的代码或数据。请检查由 -P 链接器选项分配的地址。

(599) No psect classes given for COFF write

要生成 COFF 文件，CROMWELL 要求指定程序存储器 psect 类。请确保使用 -N 选项。

(600) No chip arch given for COFF write

要生成 COFF 文件，CROMWELL 要求指定芯片架构。请确保使用 -P 选项。

(601) Unknown chip arch "*" for COFF write

CROMWELL 无法识别为生成 COFF 文件而指定的芯片架构。请确保使用 -P 选项，并且正确指定架构。

(602) null file format name

传递给 CROMWELL 的 -I 或 -O 选项必须指定文件格式。

(603) ambiguous file format name ""**

向 CROMWELL 指定的输入或输出格式具有歧义。这些格式分别使用 -i 键和 -o 键选项指定。

(604) unknown file format name ""**

向 CROMWELL 指定的输出格式是未知的，例如：

cromwell -m -P16F877 main.HEX main.sym -ocot 输出文件类型为 cot，您的本意是否是 cof？

(605) did not recognize format of input file

传递给 CROMWELL 的输入文件必须为 COD、Intel HEX、Motorola HEX、COFF、OMF51、P&E 或 HI-TECH。

(606) inconsistent symbol tables

这是内部编译器错误。详情请联系 CMS 技术支持。

(607) inconsistent line number tables

这是内部编译器错误。详情请联系 CMS 技术支持。

(608) bad path specification

这是内部编译器错误。详情请联系 CMS 技术支持。

(609) missing device spec after -P

传递给 CROMWELL 的 -p 选项必须指定器件名称。

(610) missing psect classes after -N

CROMWELL 要求为 -N 选项提供 psect 类名的列表。

(611) too many input files

指定要由 CROMWELL 转换的输入文件太多。

(612) too many output files

向 CROMWELL 指定的输出文件格式太多。

(613) no output file format specified

必须向 CROMWELL 指定输出格式。

(614) no input files specified

CROMWELL 必须具有要转换的输入文件。

(616) option -Cbaseaddr is illegal with options -R or -L

链接器选项 -Cbaseaddr 不能与 -R 或 -L 链接器选项一起使用。

(618) error reading COD file data

读取输入 COD 文件时出错。请确认在命令行上指定的文件的拼写和路径。

(619) I/O error reading symbol table

COD 文件在指定记录中具有无效的格式。

(620) filename index out of range in line number record

COD 文件在指定记录中具有无效的值。

(621) error writing ELF/DWARF section "" on ""

将指示的段写入给定文件时出错。请确认在命令行上指定的文件的拼写和路径。

(622) too many type entries

这是内部编译器错误。详情请联系 CMS 技术支持。

(623) bad class in type hashing

这是内部编译器错误。详情请联系 CMS 技术支持。

(624) bad class in type compare

这是内部编译器错误。详情请联系 CMS 技术支持。

(625) too many files in COFF file

这是内部编译器错误。详情请联系 CMS 技术支持。

(626) string lookup failed in COFF: get_string()

这是内部编译器错误。详情请联系 CMS 技术支持。

(627) missing "" in SDB file "" line * column *

这是内部编译器错误。详情请联系 CMS 技术支持。

(629) bad storage class "" in SDB file "" line * column *

这是内部编译器错误。详情请联系 CMS 技术支持。

(630) invalid syntax for prefix list in SDB file ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(631) syntax error at token "" in SDB file "" line * column *

这是内部编译器错误。详情请联系 CMS 技术支持。

(632) can't handle address size (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(633) unknown symbol class (*)

CROMWELL 在 COFF、CMS COFF 或 ICOFF 文件的符号表中遇到它无法识别的符号类。

(634) error dumping ""

CROMWELL 的输入文件为不受支持的类型，或该文件无法被转储到屏幕上。

(635) invalid HEX file "" on line *

指定的 HEX 文件包含无效的行。如果 HEX 文件是由编译器生成的，请联系 CMS 技术支持。

(636) checksum error in Intel HEX file "" on line *

在指定 Intel HEX 文件的指定行发现校验和错误。该 HEX 文件可能已损坏。

(637) unknown prefix "" in SDB file ""

这是内部编译器警告。详情请联系 CMS 技术支持。

(638) version mismatch: 0x* expected

输入 CMS COFF 文件不是使用 CROMWELL 生成的。

(639) zero bit width in CMS optional header

输入 CMS COFF 文件中的可选头部指示程序或数据存储空间为零位宽。

(668) prefix list did not match any SDB types

这是内部编译器错误。详情请联系 CMS 技术支持。

(669) prefix list matched more than one SDB type

这是内部编译器错误。详情请联系 CMS 技术支持。

(670) bad argument to -T

-T 选项的指定制表符长度的参数不存在或格式不正确。该选项要求一个十进制整型参数。

(671) argument to -T should be in range 1 to 64

-T 选项的指定制表符长度的参数不处于预期范围。该选项要求一个介于 1（含）至 64（含）范围的十进制整型参数。

(673) missing filename after * option

指示的选项需要一个有效的文件名。请确保提供给该选项的文件名参数存在，并且拼写正确。

(674) too many references to "*"

这是内部编译器错误。详情请联系 CMS 技术支持。

(677) set_fact_bit on cms17!

这是内部编译器错误。详情请联系 CMS 技术支持。

(678) case 55 on cms17!

这是内部编译器错误。详情请联系 CMS 技术支持。

(679) unknown extraspecial: *

这是内部编译器错误。详情请联系 CMS 技术支持。

(680) bad format for -P option

这是内部编译器错误。详情请联系 CMS 技术支持。

(681) bad common spec in -P option

这是内部编译器错误。详情请联系 CMS 技术支持。

(682) this architecture is not supported by the CMS C Lite compiler

指定了支持之外的目标器件。该编译器仅支持来自这些架构系列的器件。

(683) bank 1 variables are not supported by the CMS C Lite compiler

检测到绝对地址位于存储区 1 中的变量。该编译器不支持对处于该存储区中的变量进行代码生成。

(684) bank 2 and 3 variables are not supported by the CMS C Lite compiler

检测到绝对地址位于存储区 2 或 3 中的变量。该编译器不支持对处于这些存储区中的变量进行代码生成。

(685) bad putwsize()

这是内部编译器错误。详情请联系 CMS 技术支持。

(686) bad switch size (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(687) bad pushreg ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(688) bad popreg ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(689) unknown predicate ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(690) interrupt function requires address

这是内部编译器错误。详情请联系 CMS 技术支持。

(691) interrupt functions not implemented for 12 bit CMS MCU

12 位的 CMS MCU 处理器系列不支持中断。

(692) interrupt function "" may only have one interrupt level

只能有一个中断优先级与中断函数相关联。请进行检查，以确保对于指定的函数仅使用了一条 `interrupt_level pragma` 伪指令。对于从中断函数中调用的主干函数，可以多次使用该 `pragma` 伪指令。例如：

```
#pragma interrupt_level 0
#pragma interrupt_level 1 /* which is it to be: 0 or 1? */
void interrupt isr(void)
{
```

(693) interrupt level may only be 0 (default) or 1

唯一可能的中断优先级为 0 或 1。请进行检查，以确保所有 `interrupt_level pragma` 伪指令都使用这些优先级。

```
#pragma interrupt_level 2 /* oops -- only 0 or 1 */
void interrupt isr(void)
{
    /* isr code goes here */
}
```

(694) no interrupt strategy available

器件不支持在中断服务程序期间保存并随后恢复寄存器。

(695) duplicate case label (*)

在该 switch 语句中，有两个 case 标号具有相同的值，例如：

```
switch(in) {  
case '0': /* if this is case '0'... */  
    b++;  
    break;  
case '0': /* then what is this case? */  
    b--;  
    break;  
}
```

(696) out-of-range case label (*)

该 case 标号不是控制表达式可以产生的值，因而永远不会选择该标号。

(697) non-constant case label

该 switch 语句中的 case 标号具有不是常量的值。

(698) bit variables must be global or static

位变量不能为自动类型。如果需要一个作用域为代码或函数块内部的位变量，请使用 static 限定它，例如：

```
bit proc(int a)  
{  
    bit bb;          /* oops -- this should be: static bit bb; */  
    bb = (a > 66);  
    return bb;  
}
```

(699) no case labels in switch

该 switch 语句中不具有任何 case 标号，例如：

```
switch(input) {  
} /* there is nothing to match the value of input */
```

(700) truncation of enumerated value

检测到大于该编译器支持的大值的枚举值，它已被截断，例如：

```
enum { ZERO, ONE, BIG=0x99999999 } test_case;
```

(701) unreasonable matching depth

这是内部编译器错误。详情请联系 CMS 技术支持。

(702) regused(): bad arg to G

这是内部编译器错误。详情请联系 CMS 技术支持。

(703) bad GN

这是内部编译器错误。详情请联系 CMS 技术支持。

(704) bad RET_MASK

这是内部编译器错误。详情请联系 CMS 技术支持。

(705) bad which (*) after l

这是内部编译器错误。详情请联系 CMS 技术支持。

(706) bad which in expand()

这是内部编译器错误。详情请联系 CMS 技术支持。

(707) bad SX

这是内部编译器错误。详情请联系 CMS 技术支持。

(708) bad mod "+" for how = ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(709) metaregister "" can't be used directly

这是内部编译器错误。详情请联系 CMS 技术支持。

(710) bad U usage

这是内部编译器错误。详情请联系 CMS 技术支持。

(711) bad how in expand()

这是内部编译器错误。详情请联系 CMS 技术支持。

(712) can't generate code for this expression

该错误指示 C 表达式太难，代码生成器无法实际进行编译。要成功进行代码生成，代码生成器必须知道如何编译表达式，必须有足够的资源（即，寄存器或临时存储单元）可用。简化该表达式（例如，使用一个临时变量来存放中间结果）可能可以避开该消息。

关于该消息的详细信息，请联系 CMS 技术支持。

如果所编译的代码不正常，也可能会发出该错误。例如，对使用 `const` 限定的对象执行写操作的代码是非法的，将会导致警告消息，但代码生成器可能会试图生成执行写操作的代码，但不会成功。

(713) bad initialization list

这是内部编译器错误。详情请联系 CMS 技术支持。

(714) bad intermediate code

这是内部编译器错误。详情请联系 CMS 技术支持。

(715) bad pragma ""

向代码生成器传递了它无法理解的 `pragma` 伪指令。这意味着您使用的 `pragma` 伪指令并未对于目标器件实现。

(716) bad argument to -M option ""

向代码生成器传递了它无法理解的 `-M` 选项。如果它是由标准编译器驱动程序调用的，则不应发生这种情况。

(718) incompatible intermediate code version; should be *.*

由 P1 生成的中间代码文件不是可用于该代码生成器的正确版本。这或者是由于在同一目录中安装了一个或多个编译器的这种不兼容版本，或者发生了导致临时文件损坏的临时文件错误。请检查 TEMP 环境变量的设置。如果它引用很长的路径名，则将其更改为较短的路径。如果需要，请联系 CMS 技术支持了解详情。

(720) multiple free: *

这是内部编译器错误。详情请联系 CMS 技术支持。

(721) element count must be constant expression

这是内部编译器错误。详情请联系 CMS 技术支持。

(722) bad variable syntax in intermediate code

这是内部编译器错误。详情请联系 CMS 技术支持。

(723) function definitions nested too deep

C 代码不大可能发生该错误，因为 C 不能具有嵌套函数！详情请联系 CMS 技术支持。

(724) bad op (*) in revlog()

这是内部编译器错误。详情请联系 CMS 技术支持。

(726) bad op "*" in unconval()

这是内部编译器错误。详情请联系 CMS 技术支持。

(727) bad op "*" in bconfloat()

这是内部代码生成器错误。详情请联系 CMS 技术支持。

(728) bad op "*" in confloat()

这是内部编译器错误。详情请联系 CMS 技术支持。

(729) bad op "*" in conval()

这是内部编译器错误。详情请联系 CMS 技术支持。

(731) expression error with reserved word

这是内部编译器错误。详情请联系 CMS 技术支持。

(732) initialization of bit types is illegal

不能初始化 bit 类型的变量，例如：

```
bit b1 = 1; /* oops!
```

```
      b1 must be assigned after its definition */
```

(733) bad string "*" in pragma "psect"

向代码生成器传递了格式不正确的字符串的 pragma psect 伪指令，例如：

```
#pragma psect text /* redirect text psect into what? */
```

也许您的本意是以下形式的代码：

```
#pragma psect text=special_text
```

(734) too many "psect" pragmas

使用了太多的#pragma psect 伪指令。

(735) bad string "*" in pragma "stack_size"

stack_size pragma 的参数格式不正确。该 pragma 伪指令必须后跟一个数字，表示允许的大堆栈大小。

(737) unknown argument "*" to pragma "switch"

对于#pragma switch 伪指令使用了无效的 switch 代码生成方法。可能的参数包括:auto、simple 和 direct。

(739) error closing output file

在关闭文件时，编译器检测到一个错误。详情请联系 CMS 技术支持。

(740) zero dimension array is illegal

向代码生成器传递了产生零大小数组的声明。

(741) bitfield too large (* bits)

位域中的大位数为 8，长度与存储单元宽度相同。

```
struct {  
    unsigned flag : 1;  
    unsigned value : 12; /* oops -- that's larger than 8 bits wide */  
    unsigned cont : 6;  
}object;
```

(742) function "*" argument evaluation overlapped

函数调用涉及到在两个函数之间重叠的参数。以下形式的调用可能会发生这种情况：

```
void fn1(void)  
{  
    fn3( 7, fn2(3), fn2(9)); /* Offending call */  
}  
char fn2(char fred)  
{  
    return fred + fn3(5,1,0);  
}  
char fn3(char one, char two, char three)  
{  
    return one+two+three;  
}
```

其中，fn1 调用 fn3，并通过调用 fn2 对两个参数求值，该函数又反过来调用 fn3。
应修改程序结构，以防止这种类型的调用序列。

(743) divide by zero

在代码中检测到涉及被零除的表达式。

(744) static object "" has zero size

声明了静态对象，但其长度为零。

(745) nodecount = *

这是内部编译器错误。详情请联系 CMS 技术支持。

(746) object "" qualified const, but not initialized

某个对象被限定为 const 类型，但在定义时未提供初始值。由于该对象不能由 C 程序写入，这可能意味着无意中遗漏了初始值。

(747) unrecognized option "" to -Z

这是内部编译器错误。详情请联系 CMS 技术支持。

(748) variable "" may be used before set

该变量可能在赋值之前被使用。由于它是一个自动变量，这将导致它具有一个随机值，例如：

```
void main(void)
{
    int a;
    if(a)          /* oops -- a has never been assigned a value */
        process();
}
```

(749) unknown register name "" used with pragma

这是内部编译器错误。详情请联系 CMS 技术支持。

2.4 消息 750-999

(750) constant operand to || or &&

逻辑操作符||或&&的一个操作数为常量。请检查表达式是否缺少或具有错误放置的括号。如果使能了全局优化器，并且其中一个操作数是值会被代码生成器跟踪的自动或静态局部变量，则也可能出现该消息，例如：

```
{
int a;
a = 6;
if(a || b) /* a is 6, therefore this is always true */
    b++;
```

(751) arithmetic overflow in constant expression

代码生成器对某个常量表达式进行求值，产生的值对于表达式类型来说太大。触发该警告常见的代码是对有符号数据类型进行赋值。例如：

```
signed char c;
c = 0xFF;
```

作为一个有符号的 8 位量，c 只能赋值为-128 至 127。常量等于 255，超出了该范围。如果您的本意是将该变量中的所有位置 1，则使用以下形式之一：

```
c = ~0x0;
c = -1;
```

它会将变量中的所有位置 1，无论变量的长度如何，并且不会发出警告。

该警告也可能由于中间值溢出而触发。例如：

```
unsigned int i; /* assume ints are 16 bits wide */
i = 240 * 137; /* this should be okay, right? */
```

使用计算器快速检查发现 240 * 137 等于 32880，它可以轻松地存储在一个 unsigned int 中，但会产生警告。这是什么原因？因为 240 和 137 均为 signed int 值。因此，

相乘的结果也必须为 signed int 值，但 signed int 无法容纳值 32880。（两个操作数均为常量值，因此代码生成器可以在编译时对该表达式求值，但它必须遵循所有 ANSI C 规则来执行该操作。）以下代码会强制乘法运算产生无符号结果：

```
i = 240u * 137; /* force at least one operand
                to be unsigned */
```

(752) conversion to shorter data type

在该表达式中可能会发生截断，因为左值的类型比右值的类型短，例如：

```
char a;
int b, c;
a = b + c; /* int to char conversion
            can result in truncation */
```

(753) undefined shift (* bits)

尝试对值执行的移位位数大于等于该数据类型中的位数。在许多处理器上，这将产生未定义的结果。这是不可移植的代码，C 标准将其标记为具有未定义的结果，例如：

```
int input;
input <=<= 33; /* oops -- that shifts the entire value out */
```

(754) bitfield comparison out of range

这是将位域与超出位域范围的值进行比较产生的结果。即，将 2 位位域与值 5 进行比较将永远不会为真，因为 2 位位域的范围为 0 至 3。例如：

```
struct {
    unsigned mask : 2; /* mask can hold values 0 to 3 */
} value;
int compare(void)
{
    return (value.mask == 6); /* test can
}
```

(755) divide by zero

进行求值的常量表达式涉及到被零除，例如：

```
a /= 0; /* divide by 0: was this what you were intending */
```

(757) constant conditional branch

条件分支（由 if、for 和 while 等语句生成）总是遵循相同的路径。它是某种涉及变量和常量表达式的比较。要让代码生成器发出该消息，变量必须具有局部作用域（自动或静态局部变量），并且必须使能全局优化器（级别可能高于 1），并且警告级别阈值可能需要低于默认级别 0。

全局优化器会尽可能长时间地跟踪局部变量在函数期间的内容。对于将这些变量与常量进行比较的 C 代码，可以在编译时推断出比较结果，并通过对输出代码进行硬编码来避免比较，例如：

```
{
    int a, b;
    a = 5;
    /* this can never be false;      always perform the true statement */
    if(a == 5)
        b = 6;
```

会生成将 a 设置为 5，然后立即将 b 设置为 6 的代码。

对于比较 if(a==5)将不会生成任何代码。如果 a 是全局变量，其他函数（特别是中断函数）可能会对其进行修改，因而无法执行变量跟踪。

该警告可能表明编译器执行了多项优化。它可能表明，表达式缺少括号或具有错误放置的括号，导致求值产生不同于预期的值。

如果编写了诸如 while(1)形式的代码，也可能会发出该警告。要生成无限循环，请使用 for(;;)。

对于 `for` 循环会发生类似的情况，例如：

```
{
    int a, b;
    /* this loop must iterate at least once */
    for(a=0; a!=10; a++)
        b = func(a);
```

在此例中，代码生成器可以同样确定 `a` 被赋值为 0，然后立即检查以确定它是否等于 10。由于 `a` 在 `for` 循环期间会被修改，所以不能删除比较代码，但代码生成器可以调整代码，使之在循环的第一轮迭代中不执行比较；仅在随后几轮中执行比较。这可能不会减小代码的长度，但它可以加快程序执行速度。

(758) constant conditional branch: possible use of "=" instead of "=="

`if` 或其他条件构造中存在一个将常量赋予变量的表达式。这可能意味着您不小心使用了赋值 `=` 而不是比较 `==`，例如：

```
int a, b;
/* this can never be false;
   always perform the true statement */
if(a = 4)
    b = 6;
```

会将值 4 赋予 `a`，由于赋值的结果总是为 `true`，比较可以忽略，并总是对 `b` 进行赋值。您的本意是否是：

```
/* this can never be false;
   always perform the true statement */
if(a == 4)
    b = 6;
```

它会检查 `a` 是否等于 4。

(759) expression generates no code

该表达式不生成任何输出代码。请检查是否存在在函数调用中遗漏了括号之类的情况，例如：

```
int fred;
fred;    /* this is valid, but has no effect at all */
```

某些器件需要通过读取特殊功能寄存器来清零硬件标志。为了支持这种情形，在某些情况下，代码生成器会对于仅包含变量 ID 的语句生成代码。对于限定为 `volatile` 的变量可能会出现这种情形。输出代码通常将读取该变量，但不对读取的值执行任何操作。

(760) portion of expression has no effect

该表达式的一部分没有作用，对表达式的值不会产生任何影响，例如：

```
int a, b, c;
a = b,c; /* "b" has no effect,
          was that meant to be a comma? */
```

(761) sizeof yields 0

代码生成器获取了某个对象的长度，发现它为零。几乎可以肯定这表明在指针的声明中存在错误；例如，您可能声明了一个指向零长度数组的指针。通常，指向数组的指针几乎没有什么用处。如果需要一个指向未知长度的对象数组的指针，则只需要一个指向单个对象的指针，然后可以进行索引或递增。

(762) constant truncated when assigned to bitfield

常量值对于赋予它的位域结构成员太大，例如：

```
struct INPUT {
    unsigned a : 3;
    unsigned b : 5;
} input_grp;
input_grp.a = 0x12; /* oops -- 0x12 cannot fit into a 3-bit wide object */
```

(763) constant left operand to "? : " operator

条件操作符?的左操作数为常量，因此三元操作符?:的结果将总是相同的，例如：

```
a = 8 ? b : c; /* this is the same as saying a = b; */
```

(764) mismatched comparison

在某个变量或表达式和某个常量值之间进行了比较，但常量值不处于该表达式的可能值范围内，例如：

```
unsigned char c;
if(c > 300) /* oops -- how can this be true? */
    close();
```

(765) degenerate unsigned comparison

无符号值与零进行比较，它的结果将总是为 true 或 false，例如：

```
unsigned char c;
if(c >= 0)
```

将总是为 true，因为无符号值永远不会小于零。

(766) degenerate signed comparison

有符号值与该类型可能的小负值进行比较，从而比较结果将总是为 true 或 false，例如：

```
char c;
if(c >= -128)
```

将总是为 true，因为 8 位有符号 char 类型的小负值为-128。

(767) constant truncated to bitfield width

常量值对于它进行操作的位域结构成员太大，例如：

```
struct INPUT {
    unsigned a : 3;
    unsigned b : 5;
} input_grp;
input_grp.a |= 0x13; /* oops -- 0x13 too large for 3-bit wide object */
```

(768) constant relational expression

存在结果总是为 `true` 或 `false` 的关系表达式。举例来说，这可能是将无符号数与负值进行比较的结果；或将某个变量与大于它可表示的大数字的值进行比较，例如：

```
unsigned int a;  
if(a == -10)    /* if a is unsigned, how can it be -10? */  
    b = 9;
```

(769) no space for macro definition

汇编器已用尽存储器。

(772) include files nested too deep

宏扩展和包含文件处理填满了汇编器的内部堆栈。打开的宏和包含文件的大数量为 30。

(773) macro expansions nested too deep

汇编程序中的宏扩展嵌套太深。其限制为每次嵌套 30 个宏和包含文件。

(774) too many macro parameters

该宏定义中存在太多的宏参数。

(776) can't allocate space for object "" (offs: *)

汇编器已用尽存储器。

(777) can't allocate space for opnd structure within object "", (offs: *)

汇编器已用尽存储器。

(780) too many psects defined

定义了太多的 psect！好家伙，这是什么程序！

(781) can't enter abs psect

这是内部编译器错误。详情请联系 CMS 技术支持。

(782) REMSYM error

这是内部编译器错误。详情请联系 CMS 技术支持。

(783) "with" psects are cyclic

如果 Psect A 要与（with）Psect B 一起放置，Psect B 要与 Psect A 一起放置，则没有层次结构。with 标志是 psect 的一个属性，它指示该 psect 必须放在与指定 psect 相同的存储器页。

从其中一个 psect 声明中删除 with 标志。这种汇编器声明可能为以下形式：

```
psect my_text,local,class=CODE,with=basecode
```

它将定义名为 my_text 的 psect，并将它放在与 psect basecode 相同的页中。

(784) overfreed

这是内部编译器错误。详情请联系 CMS 技术支持。

(785) too many temporary labels

该汇编文件中存在太多的临时标号。汇编器多允许 2000 个临时标号。

(787) can't handle "v_rtype" of * in copyexpr

这是内部编译器错误。详情请联系 CMS 技术支持。

(788) invalid character "*" in number

数字包含不属于范围 0-9 或 0-F 的字符。

(790) end of file inside conditional

扫描“endif”以匹配先前“if”时遇到文件结束符。

(793) unterminated macro argument

宏的参数未进行终止。请注意，尖括号 (“<”) 用于括起宏参数。

(794) invalid number syntax

数字的语法无效。举例来说，它可能表示在八进制数中使用了 8 或 9，或其他格式不正确的数字。

(796) use of LOCAL outside macros is illegal

LOCAL 伪指令仅在宏内是合法的。它用于定义将对于每次宏调用都是唯一的局部标号。

(797) syntax error in LOCAL argument

在汇编器宏中使用 LOCAL 汇编器伪指令定义的某个符号的语法不正确。请确保所有符号和所有其他汇编标识符都符合目标器件的汇编语言。

(798) macro argument may not appear after LOCAL

伪指令 LOCAL 之后的标号列表不能包含该宏的任何形参，例如：

```
mmm  MACRO a1
      MOVE r0, #a1
      LOCAL a1    ; oops--
; the macro parameter cannot be used with local
ENDM
```

(799) REPT argument must be >= 0

REPT 伪指令的参数必须大于零，例如：

```
REPT -2          ; -2 copies of this code? */
  MOVE    r0, [r1]++
ENDM
```

(800) undefined symbol "***

指定符号未在该模块中定义，并且未指定 GLOBAL。

(801) range check too complex

这是内部编译器错误。详情请联系 CMS 技术支持。

(802) invalid address after END directive

在汇编器 END 伪指令之后指定的程序起始地址必须为当前文件中的一个标号。

(803) undefined temporary label

引用了一个未定义的临时标号。请注意，临时标号必须具有一个 ≥ 0 的数字。

(804) write error on object file

汇编器无法写入一个目标文件。这可能是内部编译器错误。详情请联系 CMS 技术支持。

(806) attempted to get an undefined object (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(807) attempted to set an undefined object (*)

这是内部编译器错误。详情请联系 CMS 技术支持。

(808) bad size in add_reloc()

这是内部编译器错误。详情请联系 CMS 技术支持。

(809) unknown addressing mode (*)

汇编文件中使用了未知的寻址模式。

(811) "cnt" too large (*) in display()

这是内部编译器错误。详情请联系 CMS 技术支持。

(814) device type not defined

器件必须从命令行上（如-16c84）通过器件汇编器伪指令或通过 LIST 汇编器伪指令定义。

(815) syntax error in chipinfo file at line *

chipinfo 文件的指定行中包含非标准语法。

(816) duplicate ARCH specification in chipinfo file "*" at line *

chipinfo 文件的器件段具有多个 ARCH 值。仅允许使用一个 ARCH 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(817) unknown architecture in chipinfo file at line *

读取芯片 INI 文件时遇到未知的芯片架构（系列）。

(818) duplicate BANKS for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 BANKS 值。仅允许使用一个 BANKS 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(819) duplicate ZEROREG for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 ZEROREG 值。仅允许使用一个 ZEROREG 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(820) duplicate SPAREBIT for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 SPAREBIT 值。仅允许使用一个 SPAREBIT 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(821) duplicate INTSAVE for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 INTSAVE 值。仅允许使用一个 INTSAVE 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(822) duplicate ROMSIZE for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 ROMSIZE 值。仅允许使用一个 ROMSIZE 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(823) duplicate START for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 START 值。仅允许使用一个 START 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(824) duplicate LIB for "*" in chipinfo file at line *

chipinfo 文件的器件段具有多个 LIB 值。仅允许使用一个 LIB 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(825) too many RAMBANK lines in chipinfo file for "*"

chipinfo 文件包含的器件段具有太多的 RAMBANK 字段。请减少值的数量。

(826) inverted ram bank in chipinfo file at line *

在 chipinfo 文件的 RAM 字段中指定的第二个 HEX 数必须大于第一个的值。

(827) too many COMMON lines in chipinfo file for "*"

芯片配置文件中存在太多指定公共（快速操作存储区）存储器的行。

(828) inverted common bank in chipinfo file at line *

在 chipinfo 文件的 COMMON 字段中指定的第二个 HEX 数必须大于第一个的值。如果您并未修改 chipinfo INI 文件，请联系 CMS 技术支持。

(829) unrecognized line in chipinfo file at line *

chipinfo 文件包含的器件段具有无法识别的行。如果并未编辑 INI，请联系 CMS 技术支持。

(830) missing ARCH specification for "*" in chipinfo file

chipinfo 文件的器件段不具有 ARCH 值。必须指定器件的架构。如果并未修改 chipinfo 文件，请联系 CMS 技术支持。

(832) empty chip info file "*"

chipinfo 文件不包含任何数据。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(833) no valid entries in chipinfo file

chipinfo 文件不包含任何有效的器件说明。

(834) page width must be >= 60

列表页面宽度必须至少为 60 个字符。如果小于该宽度，将无法生成格式正确的列表，例如：

LIST C=10 ; the page width will need to be wider than this

(835) form length must be >= 15

使用-Flength 选项指定的页面长度必须至少为 15 行。允许将该长度设置为零，此时会完全关闭分页。默认值为零（不分页）。

(836) no file arguments

汇编器在不带任何文件参数的情况下被调用。它无法汇编任何对象。

(839) relocation too complex

该表达式中的复杂重定位太大，无法插入目标文件。

(840) phase error

汇编器在两个不同阶段中对于某个符号计算得到不同的值。这可能是由于宏或条件汇编的诡异用法而导致的。

(841) bad source/destination for movfp/movpf instruction

使用 MOVFP/MOVPF 指令指定的绝对地址太大。

(842) bad bit number

位编号必须为介于范围 0-7 的绝对表达式。

(843) a macro name can't also be an EQU/SET symbol

发现某个 EQU 或 SET 符号与某个宏同名。这是不允许的。例如：

getval MACRO

MOV r0, r1

ENDM

getval EQU 55h ; oops -- choose a different name to the macro

(844) lexical error

在输入中发现无法识别的字符或标记。

(845) symbol "*" defined more than once

在多个位置定义了该符号。如果某个符号在同一模块中多次定义，则汇编器会发出该错误，例如：

```
_next:
```

```
    MOVE    r0, #55
```

```
    MOVE    [r1], r0
```

```
_next:                ; oops -- choose a different name
```

如果符号（C 或汇编）在不同模块中多次定义，则链接器会发出该警告。错误消息中会给出模块的名称。请注意，在编译之后，C 标识符通常会在其名称前加上下划线。

(846) relocation error

无法将两个可重定位量进行相加。可以将常量与一个可重定位值相加，而同一 psect 中的两个可重定位地址可进行相减。对于汇编器在汇编时必须知道其值的各个位置，必须使用绝对值。

(847) operand error

该操作码的操作数是无效的。关于该指令的操作数的正确形式，请查看汇编器参考手册。

(848) symbol has been declared EXTERN

某个汇编标号与一个已声明为 EXTERN 的符号使用了相同的名称。

(849) illegal instruction for this device

该器件不支持该指令。

(850) PAGESEL not usable with this device

PAGESEL 伪指令不能用于选定的器件。

(851) illegal destination

该指令的目标（,f 或,w）不正确。

(852) radix must be from 2 - 16

使用 RADIX 汇编器伪指令指定的基数必须介于 2（二进制）至 16（十六进制）的范围。

(853) invalid size for FNSIZE directive

FNSIZE 汇编器伪指令参数必须为正的常量。

(855) ORG argument must be a positive constant

ORG 汇编器伪指令的参数必须为正的常量，或者为某个设为等于正的常量的符号，例如：

```
ORG -10 /* this must a positive offset to the current psect */
```

(856) ALIGN argument must be a positive constant

align 汇编器伪指令需要一个非零的正整型参数。

(857) psect may not be local and global

局部 psect 不能与全局 psect 同名，例如：

```
psect text,class=CODE      ; text is implicitly global
```

```
    MOVE r0, r1
```

; elsewhere:

```
psect text,local,class=CODE
```

```
    MOVE r2, r4
```

如果未显式说明作用域，global 标志是 psect 的默认值。

(859) argument to C option must specify a positive constant

LIST 汇编器控制的 C=选项的参数（它设置列表输出的列宽度）必须为正的十进制常量数字，例如：

```
LIST C=a0h  ; constant must be decimal and positive,
```

```
    try: LIST C=80
```

(860) page width must be >= 49

LIST 汇编器伪指令的页面宽度子选项必须指定至少为 49 的宽度。

(861) argument to N option must specify a positive constant

LIST 汇编器控制的 N 选项的参数（它设置列表输出的页面长度）必须为正的常量数字，例如：

```
LIST N=-3  ; page length must be positive
```

(862) symbol is not external

某个符号已声明为 EXTERN，但也在当前模块中进行了定义。

(863) symbol can't be both extern and public

如果符号声明为外部符号，则需要导入它。如果它声明为公共符号，则需要从当前模块导出它。符号不能同时为这两种类型。

(864) argument to "size" psect flag must specify a positive constant

PSECT 汇编器伪指令的 size 选项的参数必须为正的常量数字，例如：

```
PSECT text,class=CODE,size=-200  ; a negative size?
```

(865) psect flag "size" redefined

PSECT 汇编器伪指令的 size 标志与先前 PSECT 伪指令不同，例如：

```
psect spdata,class=RAM,size=400
```

; elsewhere:

```
psect spdata,class=RAM,size=500
```

(866) argument to "reloc" psect flag must specify a positive constant

PSECT 汇编器伪指令的 reloc 选项的参数必须为正的常量数字，例如：

```
psect test,class=CODE,reloc=-4  ; the reloc must be positive
```

(867) psect flag "reloc" redefined

PSECT 汇编器伪指令的 `reloc` 标志与先前 PSECT 伪指令不同, 例如:

```
psect spdata,class=RAM,reloc=4
```

; elsewhere:

```
psect spdata,class=RAM,reloc=8
```

(868) argument to "delta" psect flag must specify a positive constant

PSECT 汇编器伪指令的 `DELTA` 选项的参数必须为正的常量数字, 例如:

```
PSECT text,class=CODE,delta=-2 ; negative delta value doesn't make sense
```

(869) psect flag "delta" redefined

psect 的“DELTA”选项在同一模块中多次重新定义。

(870) argument to "pad" psect flag must specify a positive constant

PSECT 汇编器伪指令的“PAD”选项的参数必须为一个非零的正整数。

(871) argument to "space" psect flag must specify a positive constant

PSECT 汇编器伪指令的 `space` 选项的参数必须为正的常量数字, 例如:

```
PSECT text,class=CODE,space=-1 ; space values start at zero
```

(872) psect flag "space" redefined

PSECT 汇编器伪指令的 `space` 标志与先前 PSECT 伪指令不同, 例如:

```
psect spdata,class=RAM,space=0
```

; elsewhere:

```
psect spdata,class=RAM,space=1
```

(873) a psect may only be in one class

不能将某个 psect 分配给多个类。该 psect 在此处的定义与其他位置的定义不同。

psect 的类通过一个标志使用以下形式指定: `psect text,class=CODE` 请查找指定了一个不同类名的其他 psect 定义。

(874) a psect may only have one "with" option

psect 只能与 (with) 一个其他 psect 一起放置。psect 的 with 选项通过一个标志使用以下形式指定:

```
psect bss,with=data
```

请查找指定了一个不同 with psect 名称的其他 psect 定义。

(875) bad character constant in expression

字符常量要求仅包含一个字符, 但发现它大于一个字符或根本无字符。特定于汇编器的示例:

```
MOV r0, #'12' ; '12' specifies two characters
```

(876) syntax error

检测到一个语法错误。它可能是由于很多原因导致的。

(877) yacc stack overflow

这是内部编译器错误。详情请联系 CMS 技术支持。

(878) -S option used: "" ignored

指定的汇编文件与-S 选项一起被传递给驱动程序。驱动程序实际上不需要执行任何操作，因为该文件已经是一个汇编文件。

(880) invalid number of parameters. Use "*" -HELP" for help

编译器驱动程序的命令行用法不正确。

(881) setup succeeded

编译器已使用--setup 驱动程序选项成功安装。

(883) setup failed

编译器未使用--setup 驱动程序选项成功安装。请确保该选项的目录参数拼写正确、对于主机操作系统语法正确，并且它存在。

(884) please ensure you have write permissions to the configuration file

驱动程序无法访问 XML 配置文件。

(889) this * compiler has expired

该编译器的演示版使用期已结束。

(890) contact CMS to purchase and re-activate this compiler

该编译器演示版安装的评估期已到期。您需要购买编译器来重新激活它。但如果您确信评估期提早结束，请联系 CMS 技术支持。

(891) can't open psect usage map file "": *

驱动程序无法打开指定的文件。在使用驱动程序选项--summary=file 时，驱动程序将生成 psect 使用情况映射文件。请确保该文件未在另一个应用程序中打开。

(892) can't open memory usage map file "": *

驱动程序无法打开指定的文件。在使用驱动程序选项--summary=file 时，驱动程序将生成存储器使用情况映射文件。请确保该文件未在另一个应用程序中打开。

(893) can't open HEX usage map file "": *

驱动程序无法打开指定的文件。在使用驱动程序选项--summary=file 时，驱动程序将生成 HEX 使用情况映射文件。请确保该文件未在另一个应用程序中打开。

(894) unknown source file type ""

无法确定指定输入文件的扩展名。驱动程序只能识别扩展名为 as、c、obj、usb、p1、lib 或 HEX 的文件。

(895) can't request and specify options in the one command

驱动程序选项--getoption 和--setoption 的用法是互斥的。

(896) no memory ranges specified for data space

未为指定器件的数据存储空间指定任何片上或外部存储器范围。

(897) no memory ranges specified for program space

未为指定器件的程序存储空间指定任何片上或外部存储器范围。

(899) can't open option file "" for application "": *

无法打开--getoption 或--setoption 驱动程序选项所指定的选项文件。如果使用--setoption 选项，请确保文件的名称拼写正确，并且文件存在。如果使用--getoption 选项，请确保可以在给定位置创建该文件，或者它不由任何其他应用程序使用。

(900) exec failed: *

无法执行所列出的子组件。文件是否存在？请尝试重新安装编译器。

(902) no chip name specified; use "*" -CHIPINFO" to see available chip names

驱动程序在未选择编译所针对的芯片的情况下被调用。使用-CHIPINFO 选项运行驱动程序将显示所有芯片的列表，可以从中选择编译所针对的芯片。

(904) illegal format specified in "" option

该选项的用法不正确。请使用-HELP 确认正确的用法，或参见介绍该选项的手册部分。

(905) illegal application specified in "" option

无法理解传递给该选项的应用程序或它不属于编译器。

(907) unknown memory space tag "" in "" option specification

该存储器选项的参数是一个字符串，但它不匹配任何有效的标记。要确定哪些标记（如果有）对于该器件是有效的。

(908) exit status = *

正在执行的子组件之一遇到问题，并返回了错误代码。该子组件应已报告了其他消息，以说明所遇到的问题。

(913) "" option may cause compiler errors in some standard header files

使用该选项将会使标准头文件中使用的一些限定符变为无效，从而导致错误。

(915) no room for arguments

代码生成器无法再分配更多的存储器。

(917) argument too long

这是内部编译器错误。详情请联系 CMS 技术支持。

(918) *: no match

这是内部编译器错误。详情请联系 CMS 技术支持。

(919) * in chipinfo file "" at line *

芯片配置文件中的指定参数是非法的。

(920) empty chipinfo file

芯片配置文件可以打开，但它为空。请尝试重新安装编译器。

(922) chip "" not present in chipinfo file ""

编译器的芯片配置文件中未出现选定的芯片。请联系 CMS，以了解是否提供对于该器件的支持，或是否需要升级编译器的版本。

(923) unknown suboption ""

该选项可以接受子选项，但无法理解该子选项。这可能只是简单的拼写错误。如果不是，请使用-HELP 来确定此处允许的子选项。

(924) missing argument to "" option

该选项要求更多的数据，但未提供任何数据。请检查该选项的用法。

(925) extraneous argument to "" option

该选项不接受附加的数据，但提供了附加的数据。请检查该选项的用法。

(926) duplicate "" option

该选项只能出现一次，但出现了多次。

(928) bad "" option value

指定的选项要求一个有效的十六进制整型参数。

(929) bad "" option ranges

该选项要求一个采用范围格式（范围起始值-范围结束值）的参数，但参数不符合该语法。

(930) bad "" option specification

该选项的参数未正确指定。使用-HELP 运行驱动程序，或参见本手册中的驱动程序章节，以验证该选项的正确用法。

(931) command file not specified

在命令行上未找到该应用程序的命令文件（要求位于“@”或“<”之后）。

(939) no file arguments

驱动程序在未在其命令行上列出输入文件的情况下被调用。如果在通过第三方 IDE 进行编译时获得该消息，则可能由于该 IDE 无法验证要编译的源文件或要链接的目标文件，因而未在命令行上提供它们。

(940) *-bit checksum * placed at *

显示所请求的校验和计算的结果。

(941) bad "" assignment; USAGE: **

HEXMATE 的一个选项的用法不正确或不完整。按照消息给出的用法操作，并确保该选项的格式正确且完整。

(942) unexpected character on line * of file ""

文件包含对于此类文件无效的字符，该文件可能已损坏。例如，Intel HEX 文件要求仅包含十六进制数字的 ASCII 表示形式、冒号 (:) 和行格式。存在任何其他字符都会导致该错误。

(944) data conflict at address *h between * and *

Hexmate 数据源请求将不同的数据存储在地址。要强制一个数据源覆盖另一个数据源，请使用“+”说明符。如果发生冲突的两个指定数据源是相同数据源，则说明数据源可能包含错误。

(945) checksum range (*h to *h) contained an indeterminate value

该校验和计算的地址范围包含一个无法解析的值。如果校验和结果要存储在地址范围内，则会发生这种情况。

(948) checksum result width must be between 1 and 4 bytes

所请求的校验和字节长度是非法的。校验和结果必须为 1 至 4 个字节宽。请检查-CKSUM 选项的参数。

(949) start of checksum range must be less than end of range

在传递给-CKSUM 选项的范围中，起始值大于结束值。参数可能不完整或输入顺序不对。

(951) start of fill range must be less than end of range

在传递给-FILL 选项的范围中，起始值大于结束值。参数可能不完整或输入顺序不对。

(953) unknown -HELP sub-option: *

传递给-HELP 的子选项是无效的。请检查子选项的拼写或使用不带子选项的-HELP 列出所有选项。

(956) -SERIAL value must be between 1 and * bytes long

所存储的序列号超出范围。请确保序列号可以存储在该选项允许的字节数内。

(958) too many input files specified; * file maximum

使用的文件参数太多。请尝试分几个阶段而不是一条命令中合并这些文件。

(960) unexpected record type (*) on line * of ""

Intel HEX 文件包含无效的记录类型。关于有效的记录类型，请参见 Intel HEX 格式规范。

(962) forced data conflict at address *h between * and *

HEXMATE 数据源强制将不同的数据存储在地址。多个使用“+”说明符的数据源在相同地址处存储数据。存储在其中的实际数据可能不是您预期的数据。

(963) checksum range includes voids or unspecified memory locations

校验和范围内的数据内容中存在间隙。由于要进行校验和计算的地址范围内存在间隙/ 未用字节，运行时计算的校验和可能会与编译时校验和不同。使用已知值填充未用单元可以更正该问题。

(964) unpaired nibble in -FILL value will be truncated

为 FILL 选项提供的十六进制代码包含不完整的字节。将忽略不完整的字节（半字节）。

(965) -STRPACK option not yet implemented, option will be ignored

该选项目前不可用，将被忽略。

(966) no END record for HEX file ""

Intel HEX 文件中不包含 END 类型的记录。该 HEX 文件可能不完整。

(967) unused function definition "" (from line *)

正在编译的模块中永远不会调用所指示的静态函数。由于为静态类型，不能从其他模块调用该函数，所以该警告意味着永远不会使用该函数。该函数或者是多余的，或者先前要调用它的代码已在编译中被排除或函数名存在拼写错误。

(968) unterminated string

字符串常量似乎缺少右引号。

(969) end of string in format specifier printf()

样式函数的格式说明符的格式不正确。

(970) character not valid at this point in format specifier printf()

样式的格式说明符中具有非法字符。

(971) type modifiers not valid with this format

类型修饰符不能用于该格式。

(972) only modifiers "h" and "l" valid with this format

只有修饰符 h（short）和 l（long）对于该 printf 格式说明符是合法的。

(973) only modifier "l" valid with this format

对于该格式唯一合法的修饰符为 l（long）。

(974) type modifier already specified

该类型中已经指定了该类型修饰符。

(975) invalid format specifier or type modifier

printf 样式字符串中的格式说明符或修饰符对于该特定格式是非法的。

(976) field width not valid at this point

在 printf() 类型格式说明符中的此处不能出现字段宽度。

(978) this identifier is already an enum tag

跟随在 `struct` 或 `union` 关键字后面的该标识符已经是某个枚举类型的标记, 因此只能跟随在关键字 `enum` 之后, 例如:

```
enum IN {ONE=1, TWO};
struct IN {           /* oops -- IN is already defined */
    int a, b;
};
```

(979) this identifier is already a struct tag

跟随在 `union` 或 `enum` 关键字后面的该标识符已经是某个结构的标记, 因此只能跟随在关键字 `struct` 之后, 例如:

```
struct IN {
    int a, b;
};
enum IN {ONE=1, TWO}; /* oops -- IN is already defined */
```

(980) this identifier is already a union tag

跟随在 `struct` 或 `enum` 关键字后面的该标识符已经是某个联合的标记, 因此只能跟随在关键字 `union` 之后, 例如:

```
union IN {
    int a, b;
};
enum IN {ONE=1, TWO}; /* oops -- IN is already defined */
```

(981) pointer required

此处需要一个指针, 例如:

```
struct DATA data;
data->a = 9;          /* data is a structure,
                      not a pointer to a structure */
```

(982) unknown op "*" in nextuse()

这是内部编译器错误。详情请联系 CMS 技术支持。

(983) storage class redeclared

某个变量先前声明 `static` 类型, 现在被重新声明为 `extern` 类型

(984) type redeclared

重新声明了该函数或对象的类型。出现该错误可能是因为存在两个不兼容的声明, 或因为隐式声明后面跟随一个不兼容的声明, 例如:

```
int a;
char a; /* oops -- what is the correct type? */
```

(985) qualifiers redeclared

该函数或变量在不同声明中具有不同的限定符。

(986) enum member redeclared

枚举的成员使用不同的值定义了两次或多次。该成员是否在同一个列表中出现两次，或者成员的名称是否在枚举列表中出现多次？

(987) arguments redeclared

传递给该函数的参数的数据类型与其原型不匹配。

(988) number of arguments redeclared

该函数声明中的参数数量与相同函数的先前声明不一致。

(989) module has code below file base of *h

该模块具有地址低于给定地址的代码，但使用 **-C** 选项指定要创建一个会映射到该地址的二进制输出文件。这意味着来自该模块的代码必须放置在该文件起始处之前！请检查汇编文件中是否缺少 **psect** 伪指令。

(990) modulus by zero in #if; zero result assumed

#if 表达式中的模运算具有零除数。结果已被假定为零，例如：

```
#define ZERO 0
#if FOO%ZERO    /* this will have an assumed result of 0 */
    #define INTERESTING
#endif
```

(991) integer expression required

在 **enum** 声明中，可以为成员赋值，但表达式必须求值为 **int** 类型的常量，例如：

```
enum {one = 1, two, about_three = 3.12};
    /* no non-int values allowed */
```

(992) can't find op

这是内部编译器错误。详情请联系 CMS 技术支持。

(993) some command-line options are disabled

编译器正在演示模式下工作。一些命令行选项会被禁止。

(994) some command-line options are disabled and compilation is delayed

编译器正在演示模式下工作。一些命令行选项会被禁止，编译速度会较慢。

(995) some command-line options are disabled, code size is limited to 16kB, compilation is delayed

编译器正在演示模式下工作。一些命令行选项会被禁止；编译速度会较慢，并且允许的大代码长度被限制为 16 KB。

2.5 消息 1000-1249

(1015) missing "*" specification in chipinfo file "*" at line *

该属性要求至少出现一次，但未为该芯片定义。

(1016) missing argument* to "*" specification in chipinfo file "*" at line *

在芯片配置文件中，该属性的该值为空。

(1017) extraneous argument* to "*" specification in chipinfo file "*" at line *

在芯片配置文件中，所列出的规范具有太多的属性。

(1018) illegal number of "*" specification* (* found; * expected) in chipinfo file "*" at line *

该属性要求出现一定次数；但是，对于该芯片它未出现。

(1019) duplicate "*" specification in chipinfo file "*" at line *

该属性只能定义一次，但对于该芯片定义了多次。

(1020) unknown attribute "*" in chipinfo file "*" at line *

芯片配置文件包含一个该版本的编译器无法理解的属性。是否芯片配置文件或驱动程序已被替换为该编译器另一个版本的等效组件？

(1021) syntax error reading "*" value in chipinfo file "*" at line *

芯片配置文件错误地为该器件定义了指定的值。如果要自己修改该文件，请小心操作，并参见该文件开始处的注释来了解关于此处要求哪种类型的值的说明。

(1022) syntax error reading "*" range in chipinfo file "*" at line *

芯片配置文件错误地为该器件定义了指定的范围。如果要自己修改该文件，请小心操作，并参见该文件开始处的注释来了解关于此处要求哪种类型的值的说明。

(1024) syntax error in chipinfo file "*" at line *

芯片配置文件中的指定行处包含语法错误。

(1025) unknown architecture in chipinfo file "*" at line *

所指示行中的属性定义了对于该编译器未知的架构。

(1026) missing architecture in chipinfo file "*" at line *

chipinfo 文件的器件段不具有 ARCH 值。必须指定器件的架构。如果并未修改 chipinfo 文件，请联系 CMS 技术支持。

(1027) activation was successful

编译器已成功激活。

(1028) activation was not successful - error code (*)

编译器未成功激活。

(1029) compiler not installed correctly - error code (*)

该编译器未找到任何激活信息，无法继续执行。编译器可能未正确或未完全安装。引用的错误代码可以帮助诊断该失败的原因。如果联系 CMS 获取有关该问题的帮助，可能会要求您提供该故障代码。

(1030) Hexmate - Intel HEX editing utility (Build 1.%i)

指示正在执行的 HEXMATE 的版本号。

(1031) USAGE: * [input1.HEX] [input2.HEX]... [inputN.HEX] [options]

HEXMATE 的建议用法。

(1032) use -HELP=<option> for usage of these command line options

通过向 HELP 选项传递某个特定选项，可以获取关于该选项的更详细信息。

(1033) available command-line options:

这是出现在该应用程序的可用选项列表之前的简单标题。

(1034) type "*" for available options

您似乎需要帮助。该建议性消息建议如何获取关于该应用程序可用选项或这些选项用法的更多信息。

(1035) bad argument count (*)

某个函数的参数数量是不合理的。这是内部编译器错误。详情请联系 CMS 技术支持。

(1036) bad "*" optional header length (0x* expected)

该 COFF 文件中的可选头部的长度不正确。

(1037) short read on *

在读取该消息中所指示类型的数据时，它在达到其指定长度之前终止。

(1038) string table length too short

COFF 字符串表的指定长度小于小值。

(1039) inconsistent symbol count

符号表中的符号数量超出了 COFF 头部中指示的数量。

(1040) bad checksum: record 0x*, checksum 0x*

指定类型的记录无法与它自己的校验和值匹配。

(1041) short record

在读取文件时，文件中的一个记录在其指定长度之前结束。

(1042) unknown * record type 0x*

该记录的类型指示符与该文件格式的所有有效类型都不匹配。

(1043) unknown optional header

读取该 CMS COFF 文件时，文件头部中的可选头部的长度不正确。

(1044) end of file encountered

在需要更多数据时发现文件结束。该输入文件是否已被截断？

(1045) short read on block of * bytes

从 UBROF 记录中读取字节数据块时，块在预期长度之前结束。

(1046) short string read

从 UBROF 记录中读取字符串时，字符串在指定长度之前结束。

(1047) bad type byte for UBROF file

该 UBROF 文件未以正确的记录开始。

(1048) bad time/date stamp

该 UBROF 文件具有错误的时间/日期标记。

(1049) wrong CRC on 0x* bytes; should be *

该 UBROF 文件中的一个结束记录具有一个不匹配的 CRC 值。

(1050) bad date in 0x52 record

该 UBROF 文件中的一个调试记录具有错误的日期组成部分。

(1051) bad date in 0x01 record

该 UBROF 文件中的程序起始记录或段记录具有错误的日期组成部分。

(1052) unknown record type

读取该 UBROF 文件时，无法确定某个记录类型。

(1053) additional RAM ranges larger than bank size

所请求的附加 RAM 块超出存储区的大小。请尝试将块拆分为不跨越存储区边界的多个范围。

(1054) additional RAM range out of bounds

通过定制 RAM 配置定义的 RAM 存储器范围超出范围。

(1055) RAM range out of bounds (*)

在芯片配置文件中或通过定制配置定义的 RAM 存储器范围超出范围。

(1056) unknown chip architecture

编译器试图针对属于不受支持或已被禁止的架构的器件进行编译。

(1057) fast double option only available on 17 series processors

不能为该器件选择快速双精度库。

(1058) assertion

这是内部编译器错误。详情请联系 CMS 技术支持。

(1059) rewrite loop

这是内部编译器错误。详情请联系 CMS 技术支持。

(1081) static initialization of persistent variable ""

持久性变量被赋予初始值。这是有些矛盾的，因为初始值将在编译器启动代码执行过程中赋予变量，但 persistent 限定符要求编译器启动代码不应更改该变量。

(1082) size of initialized array element is zero

这是内部编译器错误。详情请联系 CMS 技术支持。

(1088) function pointer "" is used but never assigned a value

进行了涉及到函数指针的函数调用，但该指针从未被赋予目标地址，例如：

```
void (*fp)(int);  
fp(23);    /* oops -- what function does fp point to? */
```

(1089) recursive function call to ""

发现对于指定函数的递归调用。调用可能是直接或间接的（使用函数指针），可能是函数调用它自身，或者是函数调用另一个函数，但另一个函数的调用图中包含指定的函数。

(1090) variable "" is not used

声明了该变量，但程序未使用它。可以考虑从程序中删除它。

(1091) main function "" not defined

未定义 main 函数。每个 C 程序都必须具有名为 main 的函数。

(1094) bad derived type

这是内部编译器错误。详情请联系 CMS 技术支持。

(1095) bad call to typeSub()

这是内部编译器错误。详情请联系 CMS 技术支持。

(1096) type should be unqualified

这是内部编译器错误。详情请联系 CMS 技术支持。

(1097) unknown type string ""

这是内部编译器错误。详情请联系 CMS 技术支持。

(1098) conflicting declarations for variable "" (*:*)

在某个变量的声明中或在某个变量的声明和定义之间检测到不同的类型信息，例如：

```
extern long int test;  
int test;    /* oops -- which is right? int or long int ? */
```

(1104) unqualified error

这是内部编译器错误。详情请联系 CMS 技术支持。

(1118) bad string "*" in getexpr (J)

这是内部编译器错误。详情请联系 CMS 技术支持。

(1119) bad string "*" in getexpr (LRN)

这是内部编译器错误。详情请联系 CMS 技术支持。

(1121) expression error

这是内部编译器错误。详情请联系 CMS 技术支持。

(1137) match() error: *

这是内部编译器错误。详情请联系 CMS 技术支持。

(1157) W register must be W9

此处所需的工作寄存器必须为 W9，但选择了其他工作寄存器。

(1159) W register must be W11

此处所需的工作寄存器必须为 W11，但选择了其他工作寄存器。

(1178) the "*" option has been removed and has no effect

该选项在该版本的编译器中不再存在，已被忽略。使用编译器的 `-help` 选项或参见手册来查找替换选项。

(1179) interrupt level for function "*" may not exceed *

指定函数的中断优先级太高。每个中断函数都会被分配一个唯一的中断优先级。在分析调用图和可重入调用函数时，将会考虑该优先级。如果使用 `interrupt_levelpragma`，请检查指定的值。

(1180) directory "*" does not exist

在安装选项中指定的目录不存在。创建目录并重试。

(1182) near variables must be global or static

使用 `near` 限定的变量必须同时使用 `static` 限定或设为全局变量。自动变量不能限定为 `near`。

(1183) invalid version number

在激活期间，未在 CMS 激活服务器数据库中找到指定序列号的匹配版本号。

(1184) activation limit reached

指定序列号的激活数量已超出许可证所允许的大数量。

(1185) invalid serial number

在激活期间，未在 CMS 激活服务器数据库中找到匹配的序列号。

(1186) license has expired

该编译器的限时许可证已到期。

(1187) invalid activation request

编译器尚未正确激活。

(1188) network error *

编译器激活软件无法通过网络连接到 CMS 激活服务器。

(1190) FAE license only - not for use in commercial applications

指示该编译器已使用 FAE 许可证激活。该许可证不允许将产品用于开发商业应用程序。

(1191) licensed for educational use only

指示该编译器已使用教育许可证激活。教育许可证仅供教育机构使用，不允许将产品用于开发商业应用程序。

(1192) licensed for evaluation purposes only

指示该编译器已使用评估许可证激活。

(1193) this license will expire on *

编译器以限时试用的形式安装。该试用将在指定日期结束。

(1195) invalid syntax for "*" option

为可接受额外参数的命令行选项提供了不适当的数据或数据不足。例如，某个选项可能要求两个均为整型的参数。将字符串作为其中一个参数传递或仅提供一个参数都可能导致该错误。

(1198) too many "*" specifications; * maximum

指定该选项的次数太多。如果可能，请尝试通过几个命令行来执行这些操作。

(1199) compiler has not been activated

该编译器的试用期已到期。编译器现在无法工作，直到使用有效序列号激活为止。请联系 CMS 购买本软件，并获取序列号。

(1200) Found %0*IXh at address *h

在该地址发现在-FIND 选项中指定的代码序列。

(1201) all FIND/REPLACE code specifications must be of equal width

该选项中的所有查找、替换和掩码属性必须为相同的字节宽度。请检查提供给该选项的参数。例如，使用 FFh（1 字节）作为掩码来查找 1234h（2 字节）会产生错误；但是，使用 00FFh（2 字节）作为掩码则可以工作。

(1202) unknown format requested in -FORMAT: *

请求了未知或不受支持的 INHX 格式。关于受支持的 INHX 格式，请参见文档。

(1203) unpaired nibble in * value will be truncated

该选项的数据未以完整字节的形式输入。数据可能不完整或遗漏了前导零。例如，值 Fh 仅包含 4 个有效数据位，不是一个完整字节。值 0Fh 包含 8 个有效数据位，它是完整字节。

(1204) * value must be between 1 and * bytes long

为该选项提供了非法长度的数据。为该选项提供的值超出了该选项所需的大值或小值界限。

(1205) using the configuration file *; you may override this with the environment variable HTC_XML

这是在编译器安装期间选择的编译器配置文件。它可以通过 HTC_XML 环境变量更改。该文件用于确定编译器的安装路径。

(1207) some of the command line options you are using are now obsolete

传递给驱动程序的一些命令行选项在该版本的编译器中现已停用；但在宽限期内，驱动程序仍然会处理这些旧选项。

(1208) use -help option or refer to the user manual for option details

检测到一个过时的选项。使用 -help 或参见手册来查找不会导致该建议性消息的替换选项。

(1209) An old CMS IDE tool suite plug-in was detected.

传递给驱动程序的选项类似于 CMS IDE 将传递给该编译器的先前版本的那些选项。其中某些选项现在已过时；但是，仍然会解释它们。

(1210) Visit the CMS website (www.CMS.com) for a possible upgrade

请访问我们的网站，查看是否有可用于解决先前编译器消息中列出的问题的升级。如果您的版本是新的，请联系 CMS 技术支持来了解进一步的信息。

(1212) Found * (%0*IXh) at address *h

在该地址发现在 -FIND 选项中指定的代码序列。

(1213) duplicate ARCH for * in chipinfo file at line *

chipinfo 文件的器件段具有多个 ARCH 值。仅允许使用一个 ARCH 值。如果您并未手动编辑 chipinfo 文件，请联系 CMS 技术支持了解详情。

(1218) can't create cross reference file *

汇编器尝试创建交叉引用文件；但是，无法创建它。请检查该文件的路径名是否正确。

(1228) unable to locate installation directory

编译器无法确定它的安装目录。

(1230) dereferencing uninitialized pointer ""

对尚未赋值的指针进行解引用。这会在运行时导致错误的行为。

(1235) unknown keyword *

无法识别 USB 描述符文件中包含的标记。

(1236) invalid argument to *: *

为可以接受额外参数的选项提供了无效的参数值。请检查该选项的用法或预期参数的语法或范围。

(1237) endpoint 0 is pre-defined

试图在 USB 文件中定义端点 0。

(1238) FNALIGN failure on *

两个函数使用 FNALIGN 伪指令对齐它们的自动/ 参数块，但一个函数会调用另一个，这意味着它们不能进行对齐。如果某个函数指针被赋予每个函数的地址，但一个函数会调用另一个函数，则会发生该错误。例如：

```
int one(int a) { return a; }
int two(int a) { return two(a)+2; } /* ! */
int (*ip)(int);
ip = one;
ip(23);
ip = two; /* ip references one and two; two calls one */
ip(67);
```

(1239) pointer * has no valid targets

进行了涉及到函数指针的函数调用，但该指针从未被赋予目标地址，例如：

```
void (*fp)(int);
fp(23); /* oops -- what function does fp point to? */
```

(1240) unknown checksum algorithm type (%i)

无法打开在-Efile 或-E+file 选项之后指定的错误文件。请进行检查，以确保文件或目录是有效的，并具有只读访问权。

(1241) bad start address in *

无法读取--CHECKSUM 选项的范围起始地址。该值必须为一个十六进制数。

(1242) bad end address in *

无法读取--CHECKSUM 选项的范围结束地址。该值必须为一个十六进制数。

(1243) bad destination address in *

无法读取--CHECKSUM 选项的目标地址。该值必须为一个十六进制数。

(1245) value greater than zero required for *

HEXMATE-FIND 选项的 align 操作数必须为正数。

(1246) no RAM defined for variable placement

未指定任何涵盖分区 RAM 存储器的存储器。

(1247) no access RAM defined for variable placement

未指定任何涵盖快速操作存储区的存储器。

(1248) symbol (*) encountered with undefined type size

代码生成器被要求定位一个变量，但该变量的长度是未知的。这是内部编译器错误。详情请联系 CMS 技术支持。

2.6 消息 1250-1499

(1250) could not find space (* byte*) for variable *

代码生成器无法在分区 RAM 中找到用于指定变量的空间。

(1253) could not find space (* byte*) for auto/param block

代码生成器无法在 RAM 中找到用于存放自动和参数变量的 psect 的空间。

(1254) could not find space (* byte*) for data block

代码生成器无法在 RAM 中找到用于存放已初始化变量的 data psect 的空间。

(1255) conflicting paths for output directory

通过-O 或--OUTDIR 选项为编译器的输出目录提供了相互矛盾的路径，例如：

```
--outdir=.././ -o../main.HEX
```

(1256) undefined symbol "" treated as HEX constant

可解释为符号或十六进制值的标记与先前定义的所有符号都不匹配，所以将被解释为后者。使用前导零来避免歧义，或使用备用的基数说明符，例如 0x。例如：

```
MOV a, F7h ; is this the symbol F7h, or the HEX number 0xF7?
```

(1257) local variable "" is used but never given a value

定义了某个自动变量并将其用在表达式中，但它未在第一次使用之前在 C 代码中赋值。自动变量不会在启动时清零，其初始值是未定义的。例如：

```
void main(void) {  
    double src, out;  
    out = sin(src); /* oops -- what value was in src? */  
}
```

(1258) possible stack overflow when calling function ""

代码生成器进行调用树分析表明硬件堆栈可能发生溢出。这应仅视为一种指示信息。中断、汇编器优化器和程序结构都可能会影响堆栈的使用情况。堆栈使用情况基于 C 程序，并不包括基于汇编代码产生的任何调用树。

(1259) can't optimize for both speed and space

为驱动程序提供了针对速度编译和针对空间编译的相互矛盾的选项，例如：

```
--opt=speed,space
```

(1260) macro "" redefined

遇到了多个同名的宏定义，例如：

```
MACRO fin
    ret
ENDM
MACRO fin    ; oops -- was this meant to be a different macro?
    reti
ENDM
```

(1261) string constant required

DS 或 DSU 伪指令需要一个字符串参数，例如：

```
DS ONE    ; oops -- did you mean DS "ONE"?
```

(1262) object "" lies outside available * space

某个绝对变量被定位到某个存储单元，但该存储单元不处于针对目标器件定义的存储器范围内，例如：

```
int data @ 0x800    /* oops -- is this the correct address? */
```

(1264) unsafe pointer conversion

指向一种结构的指针被转换为另一种结构的指针，但两种结构并不具有类似的定义，例如：

```
struct ONE {
    unsigned a;
    long b;    /* ! */
} one;
struct TWO {
    unsigned a;
    unsigned b;    /* ! */
} two;
struct ONE * oneptr;
oneptr = & two;    /* oops --
                    was ONE meant to be same struct as TWO? */
```

(1267) fixup overflow referencing * into * bytes at 0x*

更多信息，请参见错误消息 477。

(1268) fixup overflow storing 0x* in * bytes at *

更多信息，请参见错误消息 477。

(1273) Omniscient Code Generation not available in Free mode

该消息指示在该免费模式编译器下不会使能编译器的高级功能。

(1275) only functions may be qualified ""

仅在函数定义中使用时才有意义的限定符被用于变量定义。

```
interrupt int dacResult; /* oops --
```

```
the interrupt qualifier can only be used with functions */
```

(1276) buffer overflow in DWARF location list

与 ELF/DWARF 调试文件关联的缓冲区发生溢出。详情请联系 CMS 技术支持。

(1278) omitting "" which does not have a location

变量未列出存储单元，将从调试输出中省略。详情请联系 CMS 技术支持。

(1284) malformed mapfile while generating summary: CLASS expected but not found

读取以生成存储器摘要的映射文件的格式不正确。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1285) malformed mapfile while generating summary: no name at position *

读取以生成存储器摘要的映射文件的格式不正确。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1286) malformed mapfile while generating summary: no link address at position *

读取以生成存储器摘要的映射文件的格式不正确。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1287) malformed mapfile while generating summary: no load address at position *

读取以生成存储器摘要的映射文件的格式不正确。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1288) malformed mapfile while generating summary: no length at position *

读取以生成存储器摘要的映射文件的格式不正确。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1289) line range limit exceeded, debugging may be affected

某个 C 语句产生的汇编代码输出的长度超出了预设的限制。这意味着 CROMWELL 生成的调试信息可能不准确。该警告并不表明任何潜在的代码失败。

(1290) buffer overflow in DWARF debugging information entry

与 ELF/DWARF 调试文件关联的缓冲区发生溢出。详情请联系 CMS 技术支持。

(1291) bad ELF string table index

传递给 CROMWELL 的 ELF 文件的格式不正确，无法使用。(Cromwell)

(1292) malformed define in .SDB file *

传递给 CROMWELL 的指定 SDB 文件的格式不正确，无法使用。

(1293) couldn't find type for "*" in DWARF debugging information entry

无法通过传递给 CROMWELL 的 SDB 文件确定符号的类型。该文件已被编辑或已损坏，或者这是编译器错误——详情请联系 CMS 技术支持。

(1294) there is only one day left until this license expires

编译器正在以演示模式运行，评估许可证在不足一天后将到期，到期后将无法再以专业模式运行。到期之后，编译器可以在免费模式下无限期地工作，但会产生较大的二进制输出。

(1295) there are * days left until this license will expire

编译器正在以演示模式运行，在评估许可证在指定时间内到期之后，将无法再以专业模式运行。到期之后，编译器可以在免费模式下无限期地工作，但会产生较大的二进制输出。

(1296) source file "*" conflicts with "*"

编译器遇到多个具有相同基本名称的源文件。只有这些文件包含在不同目录中时，才会出现这种情况。由于编译器和 IDE 基于源文件的基本名称来设定中间文件的名称，而中间文件总是存储在相同位置，所以这种情况是非法的。请确保所有源文件的基本名称是唯一的。

(1297) option * not available in Free mode

当编译器在免费模式下工作时，某些选项不可用。被禁止的选项通常与编译器的执行方式相关（如 --GETOPTION 和 --SETOPTION），它们不会控制与代码生成相关的编译器功能。

(1298) use of * outside macros is illegal

某些汇编器伪指令（如 EXITM）只能在宏定义内使用。

(1299) non-standard modifier "*" - use "*" instead

使用了非标准的 printf 占位符修饰符。请改为使用指定的修饰符。例如，应使用标准 hh 修饰符而不是 b 来指示值应打印为 char 类型。

(1300) maximum number of program classes reached. List may be truncated

在命令行上向 CROMWELL 传递了类名的列表。如果传入的类名的数量太大，将不会使用全部类名，调试信息可能会受到影响。

(1301) invalid ELF section header. Skipping

CROMWELL 在 ELF 段头部中发现无效的段。将跳过该段。

(1302) could not find valid ELF output extension for this device

扩展名不是针对该目标器件系列的。

(1303) invalid variable location detected: * - *

无法通过 SDB 文件确定符号的位置。

(1304) unknown register name: ""

在 SDB 文件中，所指示符号的位置代表一个寄存器，但无法识别寄存器名称。

(1305) inconsistent storage class for variable: ""

无法识别 SDB 文件中所指示符号的存储类别。

(1306) inconsistent size (* vs *) for variable: ""

SDB 文件中所指示符号的长度与其类型的长度不匹配。

(1307) psect * truncated to * bytes

无法将代表堆栈或堆的 psect 设为所请求的大小，它将被截断，以装入可用的存储空间。

(1308) missing/conflicting interrupts sub-option, defaulting to ""

--INTERRUPT 选项的子选项缺失或格式不正确，例如：

--INTERRUPTS=single,multi

哎呀，您的本意是单向量还是多向量中断？

(1309) ignoring invalid runtime * sub-option (*) using default

--RUNTIME 选项的指定子选项的格式不正确，例如：

--RUNTIME=default,speed:0y1234

哎呀，它应该为 0x1234。

(1310) specified speed (*Hz) exceeds max operating frequency (*Hz), defaulting to *Hz

为--RUNTIME 选项的 perform 子选项指定的频率对于选定器件太大。

--RUNTIME=default,speed:0xffffffff

哎呀，该值太大。

(1311) missing configuration setting for config word *, using default

源代码中未提供指定字的配置设置，将使用默认值。

conflicting runtime perform sub-option and configuration word settings, assuming *Hz （驱动程序）配置设置与使用--RUNTIME 选项的 perform 子选项指定的值发生冲突，已选择了默认频率。

(1313) * sub-options ("") ignored

子选项的参数是不需要的，将被忽略。

--OUTPUT=intel:8

哎呀，:8 是不需要的。

(1314) illegal action in memory allocation

这是内部错误。详情请联系 CMS 技术支持。

(1315) undefined or empty class used to link psect *

链接器被要求将某个 psect 放置在某个类指定的地址范围内，但该类从未定义或不包含任何存储器范围。

(1316) attribute "" ignored

遇到了一个有效但解析器未实现的属性。它将被解析器忽略，该属性将没有任何作用。详情请联系 CMS 技术支持。

(1317) missing argument to attribute ""

遇到了一个需要参数的属性，但该参数不存在。详情请联系 CMS 技术支持。

(1318) invalid argument to attribute ""

遇到了属性的参数，但它的格式不正确。详情请联系 CMS 技术支持。

(1319) invalid type "" for attribute ""

这表示一个错误的选项被传递给解析器。详情请联系 CMS 技术支持。

(1320) attribute "" already exists

这表示同一属性选项被多次传递给解析器。详情请联系 CMS 技术支持。

(1321) bad attribute -T option "%s"

传递给解析器的属性选项的格式不正确。详情请联系 CMS 技术支持。

(1322) unknown qualifier "%s" given to -T

在属性选项中指定的限定符是未知的。详情请联系 CMS 技术支持。

(1323) attribute expected

使用了__attribute__伪指令，但未指定属性类型。

```
int rv (int a) __attribute__(( )) /* oops -- what is the attribute? */
```

(1324) qualifier "" ignored

一些限定符是有效的，但在一些编译器或目标器件上可能未实现。该警告指示将忽略该限定符。

(1325) no such CP* register: (\$*), select (*)

某个变量使用 cp0 限定，但在使用该变量指定的地址处不存在相应的协处理器寄存器。

```
cp0volatile unsigned int mycpvar @ 0x7000; /* oops –  
did you mean 0x700, try... */  
cp0 volatile unsigned int mycpvar @ __REGADDR(7, 0);
```

(1326) "" qualified variable (*) missing address

某个变量使用 cp0 限定，但未指定协处理器寄存器地址。

```
cp0 volatile unsigned int mycpvar; /* oops -- what address ? */
```

(1327) interrupt function "" redefined by ""

所编写的中断函数被链接到已链接了中断函数的向量位置。

```
void interrupt timer1_isr(void) @ TIMER_1_VCTR { ... }  
void interrupt timer2_isr(void) @ TIMER_1_VCTR { ... } /* oops –  
did you mean that to be TIMER_2_VCTR */
```

(1328) coprocessor * registers can't be accessed from * code

所指示指令集中的代码非法地尝试访问协处理器寄存器。请确保使用正确的指令集对相应的函数进行编码。

(1329) can only modify RAM type interrupt vectors

SETVECTOR()宏被用于尝试更改中断向量表，但该表处于 ROM 中，不能在运行时更改。

(1330) only functions or function pointers may have an instruction set architecture qualifier

指令集限定符被用于不代表可执行代码的内容。

```
mips16e int input; /* oops -- you cannot qualify a variable with an instruction set type */
```

(1331) interrupt functions may not be qualified ""

对中断函数使用了非法的函数限定符。

```
mips16e void interrupt tistr(void) @ CORE_TIMER_VCTR; /* oops --  
you cannot use mips16e with interrupt functions */
```

(1332) invalid qualifier (*) and type combination on ""

一些限定变量必须具有特定的类型或长度。检测到不允许的组合。

```
volatile cp0 int mycpvar @ __REGADDR(7,0); /* oops --  
you must use unsigned types with the cp0 qualifier */
```

(1333) can't extend instruction

尝试在指令不可扩展的位置扩展 MIPS16E 指令。这是内部错误。详情请联系 CMS 技术支持。

(1334) invalid * register operand

汇编指令使用了非法的寄存器。这是内部错误或是由手写汇编代码导致的。

```
psect my_text,isa=mips16e,reloc=4  
move t0,t1 /* oops -- these registers cannot be used in the 16-bit instruction set */
```

(1335) instruction "" is deprecated

使用了已被弃用的汇编指令。

```
beql t0,t1,12 /* oops -- this instruction is no longer supported */
```

(1336) a psect may belong to only one ISA

psect 具有一个定义所允许的指令集架构的标志。所定义 psect 的 ISA 标志与相同 psect 的另一个定义标志发生冲突。

```
mytext,global,isa=mips32r2,reloc=4,delta=1  
mytext,global,isa=mips16e,reloc=4,delta=1 /* oops -  
is this the right psect name or the wrong ISA value */
```

(1337) instruction/macro "" is not part of psect ISA

某个 psect 的 ISA 标志指定了一种架构类型，但在其中发现来自不同指令集架构的指令。

```
psect my_text,isa=mips16e,reloc=4  
mtc0 t0,t1 /* oops -- this is a 32-bit instruction */
```

(1338) operand must be a * bit value

指令的常量操作数太大，无法放入指令字段宽度。

```
psect my_text,isa=mips32r2,reloc=4
li t0,0x123456789 /* oops -- this constant is too large */
```

(1339) operand must be a * bit * value

指令的常量操作数太大，无法放入指令字段宽度，并且必须具有指示的类型。

```
addiu a3, a3, 0x123456 /* oops --
the constant operand to this MIPS16E instruction is too large */
```

(1340) operand must be >= * and <= *

操作数必须处于指定的范围内。

```
ext t0,t1,50,3 /* oops -- third operand is too large */
```

(1341) pos+size must be > 0 and <= 32

位域指令的长度和位置操作数之和必须为指定范围内的值。

```
ext t0,t1,50,3 /* oops -- 50 + 3 is too large */
```

(1342) whitespace after "\"

在反斜杠和换行符之间发现空格字符，它将被忽略。

(1343) hexfile data at address 0x* (0x*) overwritten with 0x*

指定的地址将被附加的数据覆盖。这表明有多个代码段占用相同的地址。

(1346) can't find 0x* words for psect "" in segment ""

(largest unused contiguousrange 0x%IX)

另请参见消息 (491)。新的消息形式还会指示链接器可以找到的大可用块。除非存在一个足以容纳该 psect 的单独空间，否则链接器将发出该消息。在涉及到分区或分页时，大的可用空间通常会远小于剩余的总空间。

(1347) can't find 0x* words (0x* withtotal) for psect "" in segment ""

(largest unused contiguous range 0x%IX)

另请参见消息 (593)。新的消息形式还会指示链接器可以找到的大可用块。除非存在一个足以容纳该 psect 的单独空间，否则链接器将发出该消息。在涉及到分区或分页时，大的可用空间通常会远小于剩余的总空间。

(1348) enum tag "" redefined (from *.*)

定义了多个具有相同名称的枚举标记，消息中会指示前一个定义。

```
enum VALS { ONE=1, TWO, THREE };
enum VALS { NINE=9, TEN }; /* oops -- is INPUT the right tag name? */
```

(1350) pointer operands to "-" must reference the same array

如果要将两个地址相减，地址必须属于符合 ANSI 标准的相同对象。

```
int * ip;
int fred, buf[20];
ip = &buf[0] - &fred; /* oops --
                        second operand must be an address of a "buf" element */
```

(1352) truncation of operand value (0x*) to * bits

汇编指令的操作数太大，已被截断。

```
movlw 0x321 ; oops -- is this the right value?
```

(1354) ignoring configuration setting for unimplemented word *

对于目标器件上不存在的字指定了配置字设置。

```
__CONFIG(3, 0x1234); /* config word 3 does not exist on an 18C801 */
```

(1355) inline delay argument too large

使用了内联延时序列`_delay`，但所请求的指令周期数太大。请通过多次使用该程序来实现所需的延时长度。

```
#include <cms.h>
void main(void) {
    delay(0x400000); /* oops -- cannot delay by this number of cycles */
}
```

(1356) fixup overflow referencing ** (0x*) into * byte* at 0x*/0x* -> 0x* (*/0x*)**

另请参见消息 (477)。该消息形式会预先计算引起问题的指令的地址，并将包含指令的 psect 的 delta 值考虑在内。

(1357) fixup overflow storing 0x* in * byte* at 0x*/0x* -> 0x* (*/0x*)**

另请参见消息 (477)。该消息形式会预先计算引起问题的指令的地址，并将包含指令的 psect 的 delta 值考虑在内。

(1358) no space for * temps (*)

代码生成器无法找到足以容纳该程序的临时变量（中间结果暂存变量）的空间。

(1359) no space for * parameters

代码生成器无法找到足以容纳特定函数的参数变量的空间。

(1360) no space for auto/param *

代码生成器无法找到足以容纳特定函数的自动变量的空间。一些在寄存器中传递的参数可能也需要在该自动区域中分配空间。

(1361) syntax error in configuration argument

`#pragma config` 的参数的格式不正确。

```
#pragma config WDT /* oops -- is WDT on or off? */
```

(1362) configuration setting *=* redefined

相同的 config pragma 设置被使用不同的值多次发出。

```
#pragma config WDT=OFF
```

```
#pragma config WDT=ON      /* oops -- is WDT on or off? */
```

(1363) unknown configuration setting (* = *) used

配置值和设置对于目标器件是未知的。

```
#pragma config WDR=ON      /* oops -- did you mean WDT? */
```

(1364) can't open configuration registers data file *

无法找到包含值配置设置的文件。

(1365) missing argument to pragma "varlocate"

#pragma varlocate 的参数的格式不正确。

```
#pragma varlocate      /* oops -- what do you want to locate & where? */
```

(1366) syntax error in pragma "varlocate"

#pragma varlocate 的参数的格式不正确。

```
#pragma varlocate fred   /* oops -- which bank for fred? */
```

(1367) end of file in _asm

在 _asm_endasm 块内遇到了文件结束标记。

(1368) assembler message: *

所显示的是由汇编源文件中包含的 MESSG 伪指令产生的汇编器建议性消息。

(1369) can't open proc file *

无法打开选定器件的 proc 文件。

(1370) peripheral library support is not available for the *

外设库对于选定器件不可用。

(1371) float type can't be bigger than double type; double has been changed to * bits

使用--float 和--double 选项导致 double 类型的长度小于 float 类型的长度。这是 C 标准不允许的。double 类型的长度已被增大为所指示的长度。

(1372) interrupt level cannot be greater than *

指定 interrupt_level 对于选定器件太高了。

```
#pragma interrupt_level 4
```

```
// oops - there aren't that many interrupts on this device
```

(1374) the compiler feature "" is no longer supported, *

编译器不再支持所指示的功能。

(1375) multiple interrupt functions (* and *) defined for device with only one interrupt vector

所指定的两个函数都被限定为中断函数，但目标器件仅支持一个中断向量，因而仅支持一个中断函数。

```
interrupt void isr_lo(void) {  
    // ...  
}  
interrupt void isr_hi(void) {    // oops, cannot define two ISRs  
    // ...  
}
```

(1376) initial value (*) too large for bitfield width (*)

带位域的结构被定义为使用值进行初始化。指示的值太大，无法放入相应的位域宽度。

```
struct {  
    unsigned flag :1;  
    unsigned mode :3;  
} foobar = { 1, 100 };    // oops, 100 is too large for a 3 bit object
```

(1377) no suitable strategy for this switch

编译器无法基于代码和您使用 `#pragma switch` 伪指令做出的选择，确定用于对 C `switch` 语句进行编码的 `switch` 策略。您可能需要选择不同的策略。

(1378) syntax error in pragma ""

所指定 `pragma` 伪指令的参数是无效的。

```
#pragma addrrual ingore    // oops -- did you mean ignore?
```

(1379) no suitable strategy for this switch

编译器根据一系列策略之一来编码 `switch()` 语句。`case` 分支值的具体数量和值，以及 `switch` 表达式和 `switch pragma` 将决定所选择的策略。该错误指示没有任何策略可用来编码 `switch()` 语句。请联系 CMS 支持来了解程序详情。

(1380) unable to use switch strategy ""

编译器根据一系列策略之一来编码 `switch()` 语句。`case` 分支值的具体数量和值，以及 `switch` 表达式和 `switch pragma` 将决定所选择的策略。该错误指示无法使用所请求的策略来编码 `switch()` 语句。请联系 CMS 支持来了解程序详情。

(1381) invalid case label range

为 `case` 范围提供的值不正确。它们必须构成一个升序范围，并为整型常量。`case 0 ... -2: // oops -- do you mean -2 ... 0 ?`

(1385) * "" is deprecated (declared at *:*)

代码使用的某个变量或函数已被使用属性标记为弃用。

```
char __attribute__((deprecated)) foobar;  
foobar = 9;    // oops -- this variable is near end-of-life
```

(1386) unable to determine the semantics of the configuration setting "*" for register "*"

提供给某个配置位设置的数值与数据手册中指定的设置不存在直接关联。编译器将尝试遵从您的请求，但请查看您所使用器件的数据手册。

```
#pragma config OSC=11
// oops -- there is no direct association for that value on an 18F2520
// either use OSC=3 or OSC=RC
```

(1387) inline delay argument must be constant

__delay 内联函数只能接受一个常量表达式作为其参数。

```
int delay_val = 99;
__delay(delay_val); // oops, argument must be a constant expression
```

(1388) configuration setting/register of "*" with 0x* will be truncated by 0x*

配置位被编程的值对于该设置太大或不是规定值之一。

```
#pragma config WDTPS=138 // oops -- do you mean 128?
```

(1389) attempt to reprogram configuration * "*" with * (is *)

某个已编程的配置位已被再次使用与原始设置冲突的设置进行编程。

```
#pragma config WDT=ON
#pragma config WDT=OFF // oops -- watchdog on or off?
```

(1390) identifier specifies insignificant characters beyond maximum identifier length

使用的某个标识符太长，超出所设置的标识符长度。这可能意味着可能无法正确识别这些长标识符，代码将发生失败。使用-N 选项可调整大标识符长度。

```
int theValueOfThePortAfterTheModeBitsHaveBeenSet;
// oops, make your symbol shorter or increase the maximum
// identifier length
```

(1391) constant object size of * exceeds the maximum of * for this chip

定义的 const 对象对于目标器件太大。

```
const int array[200] = { ... }; // oops -- not on a Baseline part!
```

(1392) function "*" is called indirectly from both mainline and interrupt code

主干（非中断）和中断代码都调用了某个函数。如果发出该警告，它说明此类代码当前违反了选定器件的编译器限制。

(1393) possible hardware stack overflow detected, estimated stack depth: *

编译器检测到某个程序的调用图使用的堆栈空间可能多于在目标器件上分配的空间。如果存在这种情况，代码可能会失败。编译器只能对涉及到中断时的堆栈使用情况做出假定，它们会产生堆栈使用的坏情况估计。如果发出该警告，请确认是否存在函数调用嵌套。

(1394) attempting to create memory range (* - *) larger than page size, *

编译器驱动程序检测到存储器设置包含大于器件页大小的程序存储器“页”。如果使用--ROM 选项更改了默认存储器设置，很有可能会发生这种情况。请参见您所使用器件的数据手册来确定所使用的器件的页大小，以确保使用--ROM 选项指定的所有连续存储器范围具有的边界都对应于器件页边界。

--ROM=100-1fff

以上可能需要进行分页。如果页大小为 800h，以上可指定为

--ROM=100-7ff,800-fff,1000-17ff,1800-1fff

(1395) notable code sequence candidate suitable for compiler validation suite detected (*)

编译器具有内置的检查功能，可以确定是否遇到了内部代码模板组合。如果在编译代码时发现了独特的组合，将发出该消息。该消息并不是错误或警告，显示它并不表示可能发生代码失败，但如果您愿意参与，可以将您正在编译的代码发送给技术支持，以协助进行编译器测试过程。

(1396) "*" positioned in the * memory region (0x* - 0x*) reserved by the compiler

一些存储区是保留供编译器使用的。这些区域通常不用于分配在您的代码中定义的变量。但是，通过将变量设为绝对变量，可以将变量放入这些区域，并避免链接器通常情况下会发出的错误。（绝对变量可以放在任何位置，甚至放置在其他对象之上。）来自代码生成器的该警告指示已检测到一个会被定位到编译器的保留存储器中的绝对变量。您必须将该绝对变量定位到其他位置。该消息通常在将变量放置在公共存储空间中时发出。

char shared @ 0x7; // oops, this memory is required by the compiler

(1397) unable to implement non-stack call to "*"; possible hardware stack overflow

编译器必须在不使用 CALL 汇编指令和硬件堆栈（即，使用查找表）的情况下对某个 C 函数调用进行编码，但它无法实现。如果通过指针间接调用该函数，则可能需要一条调用指令，但如果硬件堆栈已满，则额外的调用会导致堆栈溢出。

(1401) eeprom qualified variables can't be accessed from both interrupt and mainline code

所有 eeprom 变量都通过不可重入的程序进行访问。如果尝试从中断和主干代码中访问使用 eeprom 限定的变量，代码可能会发生失败。请避免在中断函数中访问 eeprom 变量。

(1402) a pointer to eeprom can't also point to other data types

指针不能同时具有处于 EEPROM 空间和普通数据空间中的目标。

(1403) pragma "*" ignored

您指定的 pragma 伪指令没有任何作用，将被编译器忽略。该消息只可能在 C18 兼容模式下发出。

#pragma varlocate "mySection" fred // oops -- not accepted

(1404) unsupported: *

unsupported__attribute__ 被用于指示某个代码功能不受支持。打印的消息将指示不受支持的功能，应避免使用它。

(1405) storage class specifier "*" ignored

您指定的存储类别是不需要的，将被编译器忽略。该消息只可能在 C18 兼容模式下发出。

```
int procInput(auto int inValue) // oops -- no need for auto
{ ...
```

(1406) auto eeprom variables are not supported

限定为 eeprom 的变量不能为自动类型。如果需要，可以定义限定为 eeprom 的静态局部对象。

```
void main(void) {
    eeprom int mode; // oops -- make this static or global
```

(1407) bit eeprom variables are not supported

限定为 eeprom 的变量不能为 bit 类型。

```
eeprom bit myEEbit; // oops -- you cannot define bits in EEPROM
```

(1408) ignoring initialization of far variables

限定为 far 类型的变量不能赋予初始值。请在后面的代码中进行赋值。

```
far int chan = 0x1234; // oops -- you cannot assign a value here
```

(1409) warning number used with pragma "warning" is invalid

用于警告 pragma 伪指令的消息编号小于零或大于可用的高消息编号。

```
#pragma warning disable 1316 13350 // oops -- possibly number 1335?
```

(1410) can't assign the result of an invalid function pointer

编译器允许一些函数通过一个强制转换为函数指针的常量进行调用，但并不是所有函数。指定的地址对于该器件无效。

```
foobar += ((int (*)(int))0x0)(77);
// oops -- you cannot call a function with a NULL pointer
```

(1411) Additional ROM range out of bounds

使用--ROM 选项指定的程序存储器位于该器件所支持的片上或外部存储器范围之外。

```
--ROM=default,+2000-2ffff
```

哎呀——存储地址太高了，是否应为 2fff？

(1412) missing argument to pragma "warning disable"

#pragma warning disable 之后应为要禁止的消息编号的逗号分隔列表。

```
#pragma warning disable // oops -- what messages are to be disabled?
```

请尝试以下形式的代码。

```
#pragma warning disable 1362
```

(1413) "*" is positioned at address 0x0 and has had its address taken; pointer comparisons may be invalid

某个绝对对象被放置在地址 0 处，并已获取其地址。根据定义，这是一个 NULL 指针，检查指针是否为 NULL（即，检查以确定地址是否有效）的代码可能会失败。

```
int foobar @ 0x00;
int * ip;
void
main(void)
{
    ip = &foobar; // oops -- 0 is not a valid address
```

(1414) option * is defunct and has no effect

不再支持所使用的选项。它将被忽略。

(1415) argument to "merge" psect flag must be 0 or 1

该 psect 标志必须赋值为 0 或 1。

```
PSECT myTxt,class=CODE,merge=true ; oops -- I think you mean merge=1
```

(1416) psect flag "merge" redefined

与先前 psect 同名的 psect 指定的 merge 标志值与先前的不同。

```
psect mytext,class=CODE,merge=1
; and later
psect mytext,class=CODE,merge=0
Oops, can mytext be merged or not?
```

(1417) argument to "split" psect flag must be 0 or 1

该 psect 标志必须赋值为 0 或 1。

```
psect mytext,class=CODE,split=5
```

哎呀，split 标志参数必须为 0 或 1。

(1418) Attempt to read "control" qualified object which is Write-Only

试图读取只写寄存器。

```
state = OPTION; // oops -- you cannot read this register
```

(1419) using the configuration file *; you may override this with the environment variable CMS_XML

这是在编译器安装期间选择的编译器配置文件。它可以通过 CMS_XML 环境变量更改。该文件用于确定编译器的安装路径。

(1420) ignoring suboption ""**

您指定的子选项在该实现中无效，将被忽略。

```
--RUNTIME=default,+ramtest
```

哎呀——什么是 ramtest?

(1421) the qualifier `__xdata` is not supported by this architecture

您指定的限定符在该实现中无效，将被忽略。

```
__xdata int coeff[2]; // that has no meaning for this target
```

(1422) the qualifier `__ydata` is not supported by this architecture

您指定的限定符在该实现中无效，将被忽略。

```
__ydata int coeff[2]; // that has no meaning for this target
```

(1423) case ranges are not supported

在 `case` 分支值中使用 GCC 样式的数值范围不符合 CCI 标准。请使用单独的 `case` 标号和值，以符合标准。`switch(input) {`

```
case 0 ... 5: // oops -- ranges of values are not supported
    low();
```

(1424) short long integer types are not supported

使用 `short long` 类型不符合 CCI 标准。请改为使用相应的 `long` 类型。

```
short long typeMod; // oops -- not a valid type for CCI
```

(1425) `__pack` qualifier only applies to structures and structure members

您指定的限定符仅在用于结构或结构成员时才有意义。它将被忽略。

```
__pack int c; // oops -- there aren't inter-member spaces to pack in an int
```

(1426) 24-bit floating point types are not supported; * have been changed to 32-bits

浮点型必须为 32 位宽，以符合 CCI 标准。这些类型将被编译为 32 位宽的量。

```
--DOUBLE=24
```

哎呀——无法设置该双精度类型长度。

(1427) machine-dependent path specified in name of included file, use `-I` instead

为符合 CCI 标准，头文件规范不能包含目录分隔符。

```
#include <inc\lcd.h> // oops -- do not indicate directories here
```

请删除路径信息，并使用 `-I` 选项来指示它，例如：

```
#include <lcd.h>
```

并发出 `-Ilcd` 选项。

(1429) attribute `""` is not understood by the compiler; this attribute will be

ignored 您使用的所指示的属性对于该实现是无效的。它将被忽略。`int x __attribute__((deprecate)) = 0;`

哎呀——您的本意是否是 `deprecated`？

(1430) section redefined from `""` to `""`

您试图将一个对象放在多个段中。

```
int __section("foo") __section("bar") myvar; // oops -- which section should it be in?
```

(1431) only variable and function definitions at file-scope may be located using __section()

不能尝试使用__section() 说明符定位局部对象。

```
int main(void) {  
    int __section("myData") counter; // oops -- you cannot specify a  
    section for autos
```

(1432) "*" is not a valid section name

使用__section() 指定的段名不是有效的段名。段名必须符合正常的 C 标识符规则。

```
int __section("28data") counter; // oops -- name cannot start with digits
```

(1433) function "*" could not be inlined

无法内联指定函数。将以正常方式调用该函数。

```
int inline getData(int port) // sorry -- no luck inlining this  
{ //...
```

(1434) missing name after pragma "intrinsic"

intrinsic pragma 需要一个函数名。大多数情况下都不需要该 pragma 伪指令。如果您的本意是对函数进行内联，请参见 inline 关键字或 pragma 伪指令。

```
#pragma intrinsic // oops -- what function is intrinsically called?
```

(1435) variable "*" is incompatible with other objects in section "

不能将具有不同启动时初始化方式的变量放入同一 psect。也即，在启动时清零的变量和赋予非零初始值的变量必须处于不同 psect。类似地，位对象不能与字节对象（如 char 或 int）混合。

```
int __section("myData") input; // okay  
int __section("myData") output; // okay  
int __section("myData") lvl = 0x12; // oops -- not with uninitialized  
bit __section("myData") mode; // oops again -- no bits with bytes  
// each different object to their own new section
```

(1436) "*" is not a valid nibble, use hexadecimal digits only

在使用__IDLOC()时，参数只能包含十六进制数字，不能包含基数说明符或其他字符。不属于十六进制数字的所有字符都会在相应单元中被编程为 0。

```
__IDLOC(0x51); // oops -- you cannot use the 0x radix modifier
```

(1437) error reading CMF (*.*): *

Cromwell 正在读取的 CMF 文件是无效的。除非您修改或生成了该文件，否则说明这是一个内部错误。详情请联系 CMS 技术支持。

(1438) pragma "*" options ignored

对 pragma 伪指令使用了不受支持的选项。选项将被忽略。

```
#pragma inline=forced // oops -- no options allowed with this pragma
```

(1439) message: *

这是一个由编程器生成的消息；有一条 `pragma` 伪指令导致了打印该建议性消息。只有在使用 IAR C 扩展时，才会打印该消息。

```
#pragma message "this is a message from your programmer"
```

(1440) big-endian storage is not supported by this compiler

您对于某个变量指定了 `__big_endian` IAR 扩展。该编译器不支持大尾数法存储格式。请删除该规范，并确保其他代码不依赖于该尾数法。

```
__big_endian int volume; // oops -- this won't be big endian
```

(1441) use __at() instead of '@' and ensure the address is applicable

在使用 IAR C 扩展时，使用了 `@ address` 说明符。所指定的任何地址很可能在新架构上是不正确的。请配合使用器件数据手册来复核该地址。要防止再次出现该警告，请改为通过 `__at()` 说明符使用经过复核的地址。

(1442) type used in definition is incomplete

在定义对象时，类型必须是完整的。如果尝试使用不完整的类型定义对象，则会发出该消息。

```
typedef struct foo foo_t; foo_t x; // oops -- you cannot use foo_t until it is fully defined
struct foo {
    int i;
};
```

(1443) unknown --EXT sub-option ""

传递给 `--EXT` 选项的子选项无效。

(1444) respecified C extension from "" to ""

`--EXT` 选项被使用多次，并具有冲突的参数。后一次使用的选项将决定编译器接受的 C 扩展。

(1445) #advisory: *

这是一个由编程人员生成的消息；有一条伪指令导致了打印该建议性消息。

```
#advisory "please listen to this good advice"
```

(1446) #info: *

这是一个由编程人员生成的消息；有一条伪指令导致了打印该建议性消息。它等同于 `#advisory` 消息（1445）。

```
#info "the following is for your information only"
```

(1447) extra -L option (-L*) ignored

该错误涉及到向预处理器传递了重复的 `-L` 选项。除非您显式地运行该应用程序，否则应将它视为内部错误。详情请联系 CMS 技术支持。

(1448) no dependency file type specified with -L option

该错误涉及到向预处理器传递了格式错误的 `-L` 选项。除非您显式地运行该应用程序，否则应将它视为内部错误。详情请联系 CMS 技术支持。

(1449) unknown dependency file type (*)

该错误涉及到向预处理器传递了未知的相依性关系文件格式。除非您显式地运行该应用程序，否则应将它视为内部错误。详情请联系 CMS 技术支持。

(1450) invalid --*-spaces argument (*)

向 Cromwell 传递的选项未与有效存储空间相关。空间参数必须为代表空间的有效编号。

`--data-spaces=a`

哦——a 不是有效的数据空间编号。

(1451) no * spaces have been defined

必须向 Cromwell 传递指示每个已编号存储空间类型的信息。这通过 `--code-spaces` 和 `--data-spaces` 选项来完成。除非您显式地运行该应用程序，否则应将它视为内部错误。详情请联系 CMS 技术支持。

(1452) one or more spaces are defined as data and code

向 Cromwell 传递的选项指示存储空间同时处于代码和数据空间。除非您显式地运行该应用程序，否则应将它视为内部错误。详情请联系 CMS 技术支持。

`--code-space=1,2 --data-space=1`

哦——空间 1 是代码还是数据？

(1453) stack size specified for non-existent * interrupt

使用了 `--STACK` 选项来指定每个堆栈的大小。大小已被用于每个中断，但编译器无法检测到相应的中断函数定义，这意味着堆栈空间永远不会被使用。请确保为器件支持的每个中断创建中断函数。

`--STACK=reentrant:20:20:auto`

您请求分配两个中断堆栈，但编译器无法检测到两个中断函数的定义。

(1454) stack size specified (*) is greater than available (*)

使用了 `--STACK` 选项来指定每个堆栈的大小，但所请求的存储器总量超出了可用的存储器量。

`--STACK=software:1000:1000:20000`

堆栈空间对于小型器件来说太大了。

(1455) unrecognized stack size "*" in ""

使用了 `--STACK` 选项来指定每个堆栈的大小，但一个或多个大小不是有效值。请仅在该选项中使用十进制数值或 `auto` 标记（对应于默认大小）。

`--STACK=software:30:all:default`

只能使用十进制数字或 `auto`。

(1456) too many stack size specifiers

在 `--STACK` 选项中指定了太多的软件堆栈大小。大堆栈大小是可选的。如果使用，请为每个中断指定一个大小，为主干代码指定一个大小。

`--STACK=reentrant:20:20:auto`

对于只有一个中断的器件，堆栈大小太多了。

(1457) local variable "" cannot be made absolute

不能指定任何局部变量的地址，不论它是自动、参数还是静态局部对象。

```
int pushState(int a) {  
int cnt __at(0x100); // oops -- you cannot specify an address ...
```

(1458) Omniscient Code Generation not available in Standard mode

该消息警告您并未使能标准工作模式下的所有优化。

(1459) peripheral library support is missing for the *

外设库没有对应于您所选器件的代码。请禁止链接外设库的选项。

(1460) function-level profiling is not available for the selected chip

请不要尝试使用函数性能分析。

(1461) insufficient h/w stack to profile function ""

函数性能分析需要一个硬件堆栈级别。整个堆栈都已被该程序使用，所以无法分析所有函数。所指示的函数不会被嵌入性能分析代码，它不会在 CMS IDE 显示的性能分析信息中产生内容。

(1462) reentrant data stack model option conflicts with stack management option and will be ignored

受管堆栈选项用于将超出硬件堆栈深度的函数调用转换为使用查找表的调用。如果还使能了可重入函数模型，则不能使能该选项。如果尝试同时使用受管堆栈和可重入函数模型选项，将生成该消息。代码将在禁止堆栈管理选项的情况下进行编译。请禁止可重入函数模型或受管堆栈选项。

(1463) reentrant data stack model not supported on this device, using compiled stack for data

目标器件不支持可重入函数。该程序将编译为将基于堆栈的数据放入编译堆栈。

(1464) number of arguments passed to function "" does not match function's prototype

使用参数调用了某个函数，但该函数的定义具有空参数列表（不同于 void 参数列表）。

```
int test();    // oops--this should define the parameters  
...  
test(12, input);
```

(1465) the stack frame size for function "" (* bytes) has exceeded the maximum allowable (* bytes)

编译器已确定所指定函数的自动、参数和临时变量的软件堆栈需求超出了允许的大值。考虑使用静态局部对象而不是自动对象。

```
reentrant int addOffset(int offset) {  
int report[400];    // oops--this will never fit on the software stack
```

(1466) registers * unavailable for code generation of this expression

编译器无法生成该语句的代码。这实际上是一个“无法生成代码”错误消息（消息 712），但无法进行编译的原因涉及没有足够的寄存器可用。关于建议的变通方法，请参见消息 712。

(1467) read-only target "*" may be indirectly written via pointer

使用了一个指向非 `const` 限定类型的指针来写入值，但编译器知道该指针的一些目标（指示了这些目标中的第一个）是使用 `const` 限定的。这可能导致由于产生其他错误消息而发生代码失败。

```
void keepTotal(char * cp) {  
    *cp += total;  
}  
char c;  
const char name[] = "blender";  
keepTotal(&c);  
keepTotal(&name[2]); // oops--will write a read-only object
```

(1468) unknown ELF/DWARF specification (*) in --output option

ELF 子选项使用了未知的标志。

—output=elf:3

不存在 `elf` 标志 3。

该 ELF 子选项及其标志通常由 CMS IDE 插件发出。如果发出了该错误，请联系 CMS 技术支持，并提供编译器和 IDE 的详细信息。

(1469) function specifier "reentrant/software" used with "*" ignored

对无法编码为使用软件堆栈的函数（已指示）使用了 `reentrant`（或 `software`）说明符。将忽略该说明符，该函数将使用编译堆栈。

```
reentrant int main(void) // oops-main cannot be reentrant
```