SAS/C® Software Diagnostic Messages

Release 6.50

SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/C Software Diagnostic Messages, Release 6.50*, Cary, NC: SAS Institute Inc., 1998. 420 pp.

SAS/C Software Diagnostic Messages, Release 6.50

ISBN 1-58025-136-6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, February 1998

Contents

Credits

Documentation

Design and Production	Design, Production, and Printing Services
Style Programming	Publications Technology Development
Technical Review	Alan R. Beale, Lawrence E. Dowty, Pat Graham, Tim Hunter, Kevin Koch, Mary Mace, Scott McElroy, Bob Patten, Jenny Ray, Thomas David Rivers, Jack J. Rouse, Jeremy Smith, Edward D. Vlazny, Richard L. Wiersma
Writing and Editing	Alan R. Beale, Patricia Glasgow Moell, Karen A. Olander, Bob Patten, David A. Teal

Software

Technical Support	Tina H. Armstrong, Ted Brooks, Gary T. Ciampa, Carl George, Scott McElroy
370 Library Development	Alan R. Beale, Lawrence E. Dowty, Russell Gonsalves, Pat Graham, Jeremy Smith
Compiler	Mary Mace, Thomas David Rivers, Edward D. Vlazny
C++	Tim Hunter, Gavin Koch, Mary Mace, Jack J. Rouse
Debugger	Lawrence E. Dowty, Russell Gonsalves, Jeremy Smith
Utilities	Lawrence E. Dowty, Tim Hunter, Ann B. Janak, Mary Mace, Jill K. Melbourne, Thomas David Rivers
Source Management & Porting	Jenny Ray, Thomas David Rivers, George W. Wilder

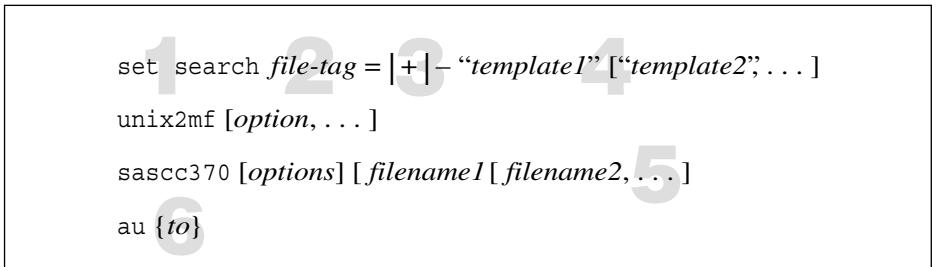
In addition, many people made development contributions to the SAS/C product in the past. They include Gary W. Black, Twilah K. Blevins, Kevin D. Bond, Oliver T. Bradley, John McDonald Brown III, Yao Chen, Mark Gass, Georges H. Guirguis, Gary H. Merrill, Bob Patten, and John A. Toebes VIII.

Using This Book

SAS/C Software Diagnostic Messages, Release 6.50 documents all SAS/C diagnostic messages generated by mainframe and workstation compilers and utility programs.

Syntax

This book uses the following syntax conventions:



1. Commands, keywords, program names, and elements of the C language appear in monospace type.
2. Values that you must supply appear in italic type.
3. Mutually exclusive choices are joined with a vertical bar(|).
4. Optional arguments appear inside square brackets ([ ]).
5. Argument groups that you can repeat are indicated by an ellipsis (. . .).
6. Abbreviations are shown by curly braces ({ }).

Additional Documentation

For a complete list of SAS publications, you should refer to the current *Publications Catalog*. The catalog is produced twice a year. You can order a free copy of the catalog by writing, calling, or faxing the Institute:

SAS Institute Inc.
Book Sales Department
SAS Campus Drive
Cary, NC 27513
Telephone: 1-800-727-3228*
Fax: 919-677-4444
E-mail: sasbook@unx.sas.com
WWW: http://www.sas.com/

* For other Institute business, call 919-677-8000.

Online Documentation

With Release 6.50, we are continuing our move to online documentation. You will find the following documentation on a CD-ROM titled *SAS/C® OnlineDoc, Release 6.50*, which is available from Book Sales.

- *Introducing SAS/C Release 6.50*
- *SAS/C Compiler and Library User's Guide, Fourth Edition, Release 6.00*
- *SAS/C C++ Development System User's Guide, Release 6.50*
- *SAS/C Software Diagnostic Messages, Release 6.50*
- *SAS/C Software: Changes and Enhancements, Release 6.50*
- *A Guide for the SAS/C Compiler Consultant, Release 6.50*
- *SAS/C Library Reference, Third Edition, Volume 1, Release 6.00*
- *SAS/C Library Reference, Third Edition, Volume 2, Release 6.00*
- *SAS/C Cross-Platform Compiler and C++ Development System: Usage and Reference, First Edition, Release 6.00*

SAS/C Software Documentation

In addition to the online documentation provided with the SAS/C software, you can order the following books:

Introducing SAS/C Software, Release 6.50 (order #A55936)
This introductory booklet is intended to give you a quick overview of SAS/C software products.

SAS/C Compiler and Library User's Guide, Fourth Edition, Release 6.00 (order #A55156)
Explains how to compile, link, and execute SAS/C programs under TSO, CMS, MVS batch, and the OpenEdition shell. This user's guide should be your starting point when learning how to use the SAS/C Compiler and Library and it will be a valuable source of information after you are familiar with the product.

SAS/C Library Reference, Third Edition, Volume 1, Release 6.00 (order #A55049)
The first of two books that comprise the authoritative reference to the SAS/C Library. Volume 1 describes all the commonly used C functions and provides a detailed overview of the SAS/C Library's organization. It also provides extensive coverage of key topics such as signal handling, input/output facilities, and environmental variables.

SAS/C Library Reference, Third Edition, Volume 2, Release 6.00 (order #A55178)
The second volume of the library reference describes all the special purpose features of the SAS/C Library such as low-level I/O and IUCV functions. This volume also provides a complete reference for the SAS/C Socket Library and the POSIX functions.

SAS/C Debugger User's Guide and Reference, Third Edition (order #A56120)
Intended for both new and experienced debugger users, this book documents the SAS/C Debugger. The guide provides and extensive tutorial and essential reference material.

SAS/C CICS User's Guide, Second Edition, Release 6.00 (order #A55117)
Provides essential information for programmers who are developing CICS applications with the SAS/C Compiler.

SAS/C Full-Screen Support Library User's Guide, Second Edition (order #A56124)
Provides all the information you will need to exploit the powerful SAS/C window interface to develop full-screen windowing applications in the IBM 3270 environment.

SAS/C Compiler Interlanguage Communication Feature User's Guide (order #A5684)
Explains how your SAS/C programs can communicate with programs written in other high-level languages such as FORTRAN, COBOL, PL/1, and Pascal.

SAS/C C++ Development System User's Guide, Release 6.50 (order #A55694)
This user's guide is the essential reference for the SAS/C C++ Development System. It explains how to compile C++ programs and also includes: detailed discussions of the basics of the product, such as header files and options; environment-specific details for using the Development System under each of the four supported environments (TSO, CMS, MVS batch, and the OpenEdition shell); information or how to debug C++ files using the SAS/C debugger.

SAS/C Cross-Platform Compiler and C++ Development System: Usage and Reference, First Edition, Release 6.00 (order #A55388)
Explains how to develop mainframe C and C++ applications from a UNIX workstation. This user's guide provides complete reference information for the SAS/C Cross-Platform Compiler and the SAS/C Cross-Platform C++ Development System.

SAS/C Software Diagnostic Messages, Release 6.50 (order #A56466)
Documents all the diagnostic messages generated by the various SAS/C software products, including error and warning messages.

SAS/C Compiler and Library Quick Reference Guide, First Edition, Release 6.00 (order #A55182)
Provides handy reference tables for compiler and linker options, debugger commands, and library functions.

SAS/C Software: Changes and Enhancements, Release 6.50 (order #A55693)
Documents all the release 6.50 changes and enhancements that have not been incorporated into books that have been updated for Release 6.50. This technical report also included the documentation for the Release 5.50 and Release 6.00 changes to the SAS/C Debugger.

A Guide for the SAS/C Compiler Consultant, Release 6.50 (order #A55937)
Provides the essential information for installing software zaps, contacting and working with the SAS/C Technical Support Department, and using the SAS/C Web page and electronic support facilities.

SAS Technical Report C-113, *SAS/C Connectivity SupportLibrary, Release 1.00* (order #A59018)
Documents the SAS/C Connectivity Support Library, which provides an interface for SAS/C programs to the X Window System, the Network File System (NFS), Remote Procedure Call (RPC), and Simple Network Management Protocol/ Distributed Program Interface (SNMP DPI).

SAS Technical Report C-115, *The Generalized Operating System Interface for the SAS/C Compiler Run-Time System, Release 5.50* (order #A59025)
Documents the Generalized Operating System (GOS) features of the SAS/C Compiler, which enable you to write interface routines for the MVS and CMS operating systems.

To order, call 1-800-727-3228. (Customers outside the United States, contact your local SAS office, or order from our Web site: **`www.sas.com/pubs.`**)

1 LSCA AR370 Messages

This chapter documents messages generated by the AR370 archive utility on mainframe and cross-platforms. After the text of each message is an explanation of the message and any variables in the message. Variables in messages are indicated by brackets [ ]. All messages can occur in any operating environment unless otherwise indicated in the explanation. In some cases, messages from the run-time library that further describe the problem may appear in conjunction with the AR370 diagnostics.

AR370 diagnostic messages have the form

```
LSCA[num] [severity]: [message text]
```

where

num	is the message number
severity	WARNING or ERROR
message text	is a description of a possible problem (WARNING) or a condition that will prevent further processing (ERROR).

Note: The AR370 message prefix LSCA has been omitted from the descriptions below.

LSCA Messages

002 **`WARNING: The continuation of control statements is not supported.`**

Explanation

MVS only: A non-blank character in column 72 of a control statement may indicate an attempt to continue the control statement on to the next card, which is not supported. The next card will be treated as a comment and is not processed. AR370 continues processing.

---

002 **`WARNING: unable to find positioning member [pos_member_name], member [member_name] appended.`**

Explanation

An attempt to add a new member or to replace or move an existing member to a position relative (BEFORE or AFTER) to an existing member of the archive failed because the positioning member specified is not currently part of the archive. AR370 continues processing. Some systems may do character translations or change the case of the command line.

Action

Make sure the spelling and case of **`[pos_mem_name]`** is correct. Use AR370 to display or type the names of the members in the archive.

---

002 `WARNING: unable to find move target [member_name].`

Explanation

An attempt to move the member **`[member_name]`** in the archive failed because a member by that name is not currently part of the archive. AR370 continues processing. Some systems may do character translations or change the case of the command line.

Action

Make sure the spelling and case of **`[member_name]`** is correct. Use AR370 to display or type the names of the members in the archive.

---

002 `WARNING: unable to find delete target [member_name].`

Explanation

An attempt to delete the member **`[member_name]`** in the archive failed because a member by that name is not currently part of the archive. Some systems may do character translations or change the case of the command line.

Action

Make sure the spelling and case of **`[member_name]`** is correct. Use AR370 to display or type the names of the members in the archive.

---

002 `WARNING: unable to find extract target [member_name].`

Explanation

An attempt to extract the member **`[member_name]`** in the archive failed because a member by that name is not currently part of the archive. Some systems may do character translations or change the case of the command line.

Action

Make sure the spelling and case of **`[member_name]`** is correct. Use AR370 to display or type the names of the members in the archive.

---

002 `WARNING: The s command modifier is not supported.`

Explanation

The s command modifier to force the regeneration of the archive symbol table is accepted for compatibility, but it has no effect. Processing continues. AR370 by default regenerates the archive symbol table whenever the archive is changed.

---

002 `WARNING: The u command modifier is not supported.`

Explanation

The u command modifier, used to prevent replacing a newer archive member with an older input file, was accepted for compatibility, but it has no effect. Processing continues.

---

002 `WARNING: The p command is not supported.`

Explanation

The p command modifier, used to print members of the archive, was accepted for compatibility, but it had no effect. Processing continues.

002 `WARNING: The f command is not supported.`

Explanation

UNIX only: The **f** command modifier, used to force truncation of file names, was accepted for compatibility, but it had no effect. Processing continues.

003 `ERROR: opening file [filename]. [message].`

Explanation

An attempt to open the file **[filename]** failed. The [message] is a brief description of the error detected by the run-time library. On MVS this error will occur when the [filename] is a ddname, and the ddname is not defined, but any file system problem or failure that might cause an open to fail could also cause this message.

004 `ERROR: reading file [filename]. [message].`

Explanation

An error occurred when attempting to read from the file or archive named **[filename]**. The **[message]** is a brief description of the error detected by the run-time library.

This diagnostic may be produced if the archive has been modified by any utility other than AR370, but any file system problem or failure that might cause a read to fail could also cause this message.

Action

Re-run AR370 with the =warning run-time option to provide additional SAS/C Library diagnostics. Check all input files for validity and integrity. In general input files should be fixed length record format with 80 byte records.

005 `ERROR: Not enough memory available, AR370 cannot continue.`

Explanation

An attempt to allocate memory failed.

Action

Try to increase the amount of memory available to AR370.

006 `ERROR: positioning file [filename]. [message].`

Explanation

An attempt to move read/write file pointer within an archive named `[filename]` failed. The `[message]` is a brief description of the error detected by the run-time library.

This diagnostic may be produced if the archive has been modified by any utility other than AR370, but any file system problem or failure that might cause a seek to fail could also be the cause.

Action

Re-run AR370 with the =warning run-time option to provide additional SAS/C Library diagnostics. Check the file for validity and integrity.

---

007 `ERROR: writing file [filename]. [message].`

Explanation

An attempt to write one or more items to the output file stream was unsuccessful. Usually this is caused by having insufficient space available for all the output, but any file system problem or failure that might cause a write to fail could also be the cause.

Action

Make sure the space available for the output file is large enough to hold all the output. Re-running AR370 with the =warning run-time option may provide additional SAS/C Library diagnostics.

---

008 `ERROR: Unable to replace original archive [Olibname] with the modified archive [Mlibname].`

Explanation

An attempt to replace the original archive [Olibname] with the new or modified archive [Mlibname] failed. The original archive may be in an inconsistent or corrupted state. On some systems it may be possible to recover the members of the old archive from the modified archive.

A shortage of disk space is the most common cause for this failure but any file system problem or failure that might cause a rename or a close to fail could also be the cause.

---

010 `ERROR: archive format unrecognized. Can not process file [filename].`

Explanation

A file `[filename]` specified as an archive did not contain a valid archive. Data read from the file was checked to verify it was an archive. If the archive has been modified by any utility other than AR370, then data could be lost or corrupted.

---

011 **`ERROR: invalid file name, [filename]. Too many or misplaced parentheses.`**

Explanation

An input file name **`[filename]`** was found to contain parentheses that were either out of order or repeated in the same name. Make sure only one set of parentheses is used in the name. The replace-as operator '=' could also cause this error if used incorrectly or as part of a name.

011 **`ERROR: invalid file name, [filename]. No usable characters found in file name.`**

Explanation

An input file name **`[filename]`**, was found to contain characters that were only delimiters or operators. Blanks and parentheses and the replace-as operator '=' are delimiters. The replace-as operator '=' could also cause this error if used incorrectly or as part of a name.

011 **`ERROR: invalid file name, [filename]. No usable characters found in member name.`**

Explanation

An attempt to derive the archive member name or basename for the input file name **`[filename]`** found characters that were only delimiters or operators in the archive member name or basename. Blanks and parentheses and the replace-as operator '=' are delimiters. The replace-as operator '=' could also cause this error if used incorrectly or as part of a name.

011 **`ERROR: invalid file name, [string] is not recognized as a file identifier.`**

Explanation

An attempt to process a command-line parameter **`[string]`** as a file identifier failed.

Action

Check the parameter **`[string]`** to make sure it is in the correct position and correctly specified.

011 **`ERROR: invalid file name, [filename]. The replace-as operator requires a fully specified file identifier.`**

Explanation

On CMS only: An incomplete file identifier was used with the replace-as operator. All input files to be added or replaced with an alternate member name must be completely specified. No defaults or wild cards can be used.

012 `ERROR: invalid INCLUDE card in SYSIN. Card follows: [card_text].`

Explanation

A card read from the data set allocated to SYSIN did not contain a valid INCLUDE control statement, a blank line, or a comment (first character '*'). The contents of the card, [card_text], are echoed on the line after this message.

Action

Make sure all INCLUDE statements conform to the general form and syntax of linkage editor control statements.

---

013 `ERROR: invalid SYMDEF card in file [filename].`

Explanation

An AR370 SYMDEF control statement in the input file [filename] contained invalid syntax. Other AR370 diagnostics that give more information on the error may precede this message.

Action

Check the SYMDEF control statement in the specified input file to make sure it conforms to the general form and syntax of linkage editor control statements. Make sure the symbol names are between 1 and 8 characters in length.

---

014 `ERROR: invalid name for DD name [ddname].`

Explanation

On MVS only: the DD name `[ddname]` from an INCLUDE statement was too long. Check the INCLUDE statements in SYSIN to make sure they conform to the general format of linkage editor control statements and are properly delimited.

---

014 `ERROR: invalid name [altname] specified as the alternate name for the DD name [ddname], must be between 1 and 18 characters in length.`

Explanation

The alternate archive member name `[altname]` for the specified input `[ddname]` was too long. All alternate member names must be at least 1 character and no more than 18 characters in length.

Action

Make sure the replace-as operator = is in the correct position and check the SYSIN, INCLUDE control statements to make sure they conform to the general format of linkage editor control statements and are properly delimited.

---

014 **`ERROR: invalid name for symbol [symbol] specified in a SYMDEF control statement. SYMDEF symbols must be 1 to 8 characters in length.`**

Explanation

The symbol name **`[symbol]`** was too long. Symbols specified via SYMDEF control statements must be at least 1 character and not more than 8 characters in length.

Action

Check the symdef cards in the input object files.

---

014 **`ERROR: invalid name in replace-as operation [input_string]. Too many replace as operators used.`**

Explanation

More than one replace-as operator = was used when specifying the **`input_string`**. **`[input_string]`** is the target file name and its alternate archive member name. Archive member names that contain the replace-as operator **`=`** cannot be created.

---

015 **`ERROR: no members in archive.`**

Explanation

This message and a zero return code were the result of an attempt to display or type all members of an empty archive. If target members were specified, a nonzero return code will result.

2 LSCC Compiler Messages

This chapter describes diagnostic messages for the SAS/C mainframe Compiler and SAS/C Cross-Platform Compiler. Most of the Cross-Platform Compiler messages, which are generated on a workstation running under UNIX, are identical to those generated by the SAS/C Compiler on the mainframe.

Diagnostic messages include numbered and unnumbered messages that are produced during the three phases of the compiler. All lexical and syntax diagnostics are produced by phase 1 of the compiler, and code generation diagnostics are produced by phase 2. Messages containing the text CXERR are compiler internal errors and are not described in this chapter.

Compiler Message Processing

The SAS/C mainframe Compiler provides a return code indicating the overall success of the compilation. For the SAS/C mainframe Compiler, a return code of 0 indicates successful compilation, a return code of 4 indicates a warning level diagnostic, and a return code equal to or greater than 8 indicates one or more error conditions were detected. Global optimization warnings do not change the return code.

The Cross-Platform Compiler return codes, unlike those of the mainframe compiler, have been chosen to be compatible with traditional UNIX program development tools such as **`make`**. If the return code is a nonzero integer, an error occurred; if it is zero, no error occurred. Cross-platform warning messages do not affect the return code. If the **`-mrc`** option is specified, the return code is the same as would be found on the mainframe.

Message levels ERROR and WARNING can be controlled by compiler options. For example, the **`enforce`** (SAS/C mainframe Compiler) option or **`-w~n`** (SAS/C Cross-Platform Compiler) option can be used to turn any WARNING into an ERROR. Similarly, the **`suppress`** or **`-wn`** option can be used to make any phase 1 WARNING disappear. This chapter describes the default message level (that is, no options affect the message level).

Message Types

Compiler error messages can be either numbered or unnumbered. Numbered messages can be produced throughout compilation. Unnumbered messages generally refer to errors occurring during compiler option processing. Unnumbered messages appear at the end of this chapter.

When the compiler detects an error in an input file, it generates a diagnostic for numbered messages in the following form:

```
[message num]  Error in [file]  at line [line num]
               before column [col num] : [text]
```

where

message num — is the message identifier number.

file — is the name of the current input file, which may be the name of a **`#include`** file.

line num — is the line number of the current line in the file.

col num — is the column where the error was detected. A vertical bar (|) can be placed in that column.

text — is a description of the condition responsible for producing the message.

Sometimes recovery from a single error may produce a succession of subsequent errors in a cascade effect. You should attempt to correct the obvious error or errors first and not be overly concerned about subsequent error messages for apparently valid source lines.

Message Numbering

Messages are numbered as follows:

- 001 through 199 and 400 through 499 are generated by phase 1.
- 200 through 299 are generated by phase 2.
- 300 through 399 are generated by the global optimization phase.

Phase 1 Error and Warning Messages

Messages 001 through 199 and 400 through 499 describe syntax or specification errors or warnings in the source file. A number of messages may be issued as warnings under some conditions and as errors under other conditions. In such cases, the distinction between these conditions is documented. Warning messages generally indicate the use of non-portable or dubious C constructs, but do not prevent generation of object code. When warnings are generated, it is advisable to verify that the actual effect of the construct diagnosed is what was intended.

Phase 2 Error and Warning Messages

The second phase of the compiler generates diagnostics 200 through 299. These messages describe conditions encountered in the process of code generation. The **`suppress`** or **`-w`***n* option cannot be used to ignore phase 2 error and warning messages.

Global Optimization Phase Error and Warning Messages

The global optimization phase generates diagnostics 300 through 399. For messages 301 and 305, examine the program source code to determine if the program is performing as intended. As with phase 2 errors, the **`suppress`** or **`-w`***n* option cannot be used to ignore these error and warning messages.

Note: The LSCC prefix that appears when a diagnostic message is generated has been omitted in the following descriptions. Additionally, SAS/C Compiler and SAS/C Cross-Platform messages are not always identical, so the exceptions are noted in the explanatory text.

LSCC Messages

001 **`ERROR/WARNING: Invalid preprocessor command.`**

Explanation

This message was generated by a variety of errors in preprocessor commands, including specifying an unrecognized command, failing to include white space between command elements, or using an illegal preprocessor symbol. The warning is generated when the **`sizeof`** keyword or any cast expression in a **`#if`** command is used. Such uses are non-ISO/ANSI.

---

002 **`ERROR: Unexpected end of file.`**

Explanation

The end of an input file was encountered when the compiler expected more data. This error may occur in a **#include** file or the original source file. In many cases, correcting a previous error eliminates this error.

003 **ERROR: File not found [filename].**

Explanation

The filename specified in a **#include** command was not found.

004 **ERROR: Invalid lexical token.**

Explanation

An unrecognized element was encountered in the input file that cannot be classified as any of the valid lexical constructs (such as an identifier or one of the valid expression operators). This error may occur if control characters or other illegal characters are detected in the source file.

005 **ERROR: Invalid macro usage: [macro]**

Explanation

A preprocessor **#define** macro was used with the wrong number of arguments.

006 **ERROR: Buffer overflow:expansion buffer.**

Explanation

Expansion of a **#define** macro caused the compiler's expansion buffer to overflow. This error may occur if more than one lengthy macro appears on a single input line, or if a macro call is ill-formed by having too few right parentheses. In the following example, if **MAC** is a macro taking two arguments and a right parenthesis is omitted in its call, the preprocessor will continue past the semicolon, believing that it is still gathering arguments for the macro:

```
MAC(f(1),f(2);
```

Although there is no fixed buffer size, an expansion that requires more space than the longest logical line occurring thus far in the compilation may cause buffer overflow. This error rarely occurs.

007 **ERROR: #include nesting level exceeded.**

Explanation

The maximum extent of **#include** file nesting was exceeded; the compiler supports **#include** nesting to a maximum depth of 16.

008 `ERROR: Invalid conversion.`

Explanation

An invalid arithmetic or pointer conversion was specified. This error usually occurs when an attempt is made to convert something into an array, a structure, or a function.

009 `ERROR: Undefined identifier [symbol].`

Explanation

The named identifier had not been previously declared. This error is generated only once; subsequent encounters with the identifier assume that it is of type `int` (which may cause other errors).

010 `ERROR: Invalid subscript expression.`

Explanation

An error was detected in the expression following the left bracket character (presumably a subscript expression). This error may occur if the expression in brackets is null (not present).

011 `ERROR: String too large or not terminated.`

Explanation

If the closing double quote was omitted in specifying a string, the string is considered not to be terminated. There is no compiler-imposed limit on the size of a string. A string may be too large only if its size exceeds available memory.

012 `ERROR: Invalid structure reference.`

Explanation

The expression preceding the period (`.`) or (→) structure reference operator was not recognized by the compiler as a structure or pointer to a structure.

013 `ERROR: Member name missing.`

Explanation

An identifier indicating the desired aggregate member was not found following the period (`.`) or (→) operator.

014 `ERROR: Undefined member [symbol]`.

Explanation

The indicated identifier was not a member of the structure or union to which the period (`.`) or (→) referred.

015 **`ERROR: Invalid function call.`**

Explanation

The identifier preceding the `(` function call operator was not implicitly or explicitly declared as a function.

016 **`ERROR: Invalid function argument.`**

Explanation

An invalid function argument expression followed the `(` function call operator. This error may occur if an argument expression is omitted as in the following:

```
f(;
f(.);
f(,1);
```

017 **`ERROR: Too many operands.`**

Explanation

More than one operand was still awaiting evaluation when the compiler encountered the end of an expression. This error may occur if an expression contains an incorrectly specified operation.

018 **`ERROR: Unresolved operator.`**

Explanation

An operator was still awaiting evaluation when the compiler encountered the end of an expression. This message may occur if an operand is omitted for a binary operation.

019 **`ERROR: Unbalanced parentheses.`**

Explanation

The number of opening and closing parentheses in an expression was not equal. This error message may also occur if a macro is poorly specified or improperly used.

020 **`ERROR: Invalid constant expression.`**

Explanation

The compiler required a constant result but the expression did not evaluate to a constant. This error may occur if one of the operators present is not valid for constant expressions.

021 `ERROR: Illegal use of aggregate.`

Explanation

An identifier declared as a structure or union was used in a context requiring a non-aggregate.

022 `WARNING: Floating point value assigned to shorter type.`

Explanation

A floating-point value (float or double) was assigned to a shorter type, and precision could be compromised. The message is suppressed unless the **`enforce`** option (or cross-platform **`-w~`***n* option) is specified.

023 `ERROR: Invalid use of conditional operator.`

Explanation

The conditional operator was used erroneously. This error may occur if the question mark (**`?`**) operator is present, but the colon (**`:`**) is not found where expected.

024 `ERROR: Pointer operand required.`

Explanation

The context of the expression required an operand be a pointer. This error may occur if the expression following does not evaluate to a pointer.

025 `ERROR: Modifiable lvalue required.`

Explanation

The context of the expression required that an operand be a modifiable lvalue. A modifiable lvalue is an expression that designates an object in memory that can be altered. For example, a simple variable (not declared **`const`**) is a modifiable lvalue. This error may occur if the expression following **`&`** is not a modifiable lvalue or if the left side of an assignment expression is not a modifiable lvalue. Note that a cast does not yield a modifiable lvalue.

026 `ERROR: Arithmetic operand required.`

Explanation

The context of the expression required that the operand be arithmetic (not a pointer, function, or aggregate).

027 `ERROR: Arithmetic or pointer operand required.`

Explanation

The context of the expression required that the operand be either arithmetic or a pointer. This error may occur for the logical OR and logical AND operators.

028 **`ERROR: Missing operand.`**

Explanation

An insufficient number of operands were available for evaluation by the time the end of an expression was encountered. This error may occur if a binary operation is improperly specified.

029 **`ERROR: Invalid pointer operation.`**

Explanation

An operation (other than addition or subtraction) was specified that was invalid for pointer operands. The compiler also detects attempts made to operate on a pointer to a structure, union, or enumeration whose tag has not been defined, as in the following code where the tag **`X`** was never defined:

```
struct X *p;
.
.
.
p++;
```

030 **`WARNING: Pointers do not point to same type of object.`**

Explanation

In an assignment statement defining a value for a pointer variable, the expression on the right side of the equal (**`=`**) operator did not evaluate to a pointer of the same type as the assigned or NULLpointer variable. The warning also occurs when a pointer of any type is assigned to an arithmetic object.

The same message is generated as an **`ERROR`** when an expression initializes a pointer to an **`int`** (or vice versa), or during a pointer comparison operation.

031 **`ERROR: Integral operand required.`**

Explanation

In the context of this expression, an operand must be an integer type. Valid integer types are **`char`**, **`int`**, **`short`**, **`long`**, **`unsigned char`**, **`unsigned int`**, **`unsigned short`**, or **`unsigned long`**.

032 **`ERROR: Invalid conversion specified.`**

Explanation

An expression was specified requesting a conversion from one data type to another data type for which there is no conversion rule.

033 ERROR: Illegal initializer expression.

Explanation

An attempt was made to initialize a variable that cannot be initialized. Also, a function cannot be initialized. An initialization cannot be part of the declaration of an **extern** variable that has been declared in function (or inner block) scope, as this example illustrates:

```
func()


extern int count = 0;
.
.
.
```

034 ERROR: Invalid initializer expression.

Explanation

An invalid expression attempted to initialize an object. This error may occur for a variety of reasons, including failure to separate elements in an initializer list with commas or to initialize an array to a single object. The following expressions are invalid:

```
char a [10] = 0;
char b [3] = 1 2 3;
```

035 ERROR: Closing brace expected.

Explanation

The compiler expected but did not find a closing right brace while processing an initializer list, structure, or union member declaration list. This error may occur if too many elements are specified in an initializer expression list or if a structure member is improperly declared.

036 WARNING: Control cannot reach this statement.

Explanation

The compiler could not reach, and thus execute, the code in a statement. Examples of possible causes include: a preceding statement such as **return**, **continue**, or **goto** precipitating an unconditional transfer of control; code within a switch block that was not reachable via a case or default statement.

037 `ERROR: Duplicate statement label [symbol].`

Explanation

The specified statement label was encountered more than once while the current function was processing.

038 `ERROR: Unbalanced braces.`

Explanation

In a body of compound statements, the number of opening left braces ({) and closing right braces (}) was not equal. The compiler may have been out of phase because of a previous error.

039 `ERROR: Invalid use of keyword [symbol].`

Explanation

One of the C language reserved words appeared in an invalid context (for example, as a variable name).

040 `ERROR: Break not inside loop or switch.`

Explanation

A **`break`** statement was detected that was not within the scope of a **`while`**, **`do`**, **`for`**, or **`switch`** statement. This error may be caused by an error in a preceding statement.

041 `ERROR: Case not inside switch.`

Explanation

A **`case`** prefix was encountered outside the scope of a **`switch`** statement. This error may be caused by an error in a preceding statement.

042 `ERROR: Invalid case expression.`

Explanation

The expression defining a **`case`** value did not evaluate to an integral constant.

043 `ERROR: Duplicate case value.`

Explanation

A **`case`** prefix was encountered that defined a constant value already used in a previous **`case`** prefix within the same **`switch`** statement.

044 `ERROR: Continue not inside loop.`

Explanation

A **`continue`** statement was detected that was not within the scope of a **`while`**, **`do`**, or **`for`** loop. This message may be caused by an error in a preceding statement.

045 `ERROR: Default not inside switch.`

Explanation

A **`default`** prefix was encountered outside the scope of a **`switch`** statement. This message may be caused by an error in a preceding statement.

046 `ERROR: More than one default.`

Explanation

The compiler encountered a **`default`** prefix within the scope of a **`switch`** statement that already contained a **`default`** prefix.

047 `ERROR: While missing from do statement.`

Explanation

The **`while`** clause was expected but not found after the body of a **`do`** statement. This message may be caused by an error within the body of the **`do`** statement.

048 `ERROR: Invalid while expression.`

Explanation

The expression defining the looping condition in a **`while`** or **`do`** loop was NULL (not present). Indefinite loops must supply a constant, such as 1, if that is what is intended.

049 `ERROR: Else not associated with if.`

Explanation

An **`else`** keyword was detected that was not within the scope of a preceding **`if`** statement. This message may be caused by an error in a preceding statement.

050 `ERROR: label missing from goto.`

Explanation

A statement label following the **`goto`** keyword was expected but not found. This message may be caused by an error in a preceding statement.

052 `ERROR: Invalid if expression.`

Explanation

The expression following the **`if`** keyword was not present.

053 `ERROR: Invalid return expression.`

Explanation

The expression following the **`return`** keyword could not be converted to the type of the value returned by the function.

054 `ERROR: Invalid switch expression.`

Explanation

The expression defining the value for a **`switch`** statement could not be converted to an integral type.

055 `WARNING: No case values for switch statement.`

Explanation

The statement defining the body of a **`switch`** statement did not contain at least one **`case`** prefix.

056 `ERROR: Colon expected.`

Explanation

The compiler expected a colon (**`:`**), but did not find one. This error may be generated if a **`case`** expression is improperly specified or if the colon was omitted following a label or prefix to a statement.

057 `ERROR: Semi-colon expected.`

Explanation

The compiler expected but did not find a semicolon (**`;`**). Typically, the compiler completed the processing of a statement but did not find the statement terminator. This error may be caused by too many closing parentheses, too few closing braces, or a statement that is otherwise formed incorrectly.

058 `ERROR: Missing parentheses.`

Explanation

A parenthesis required by the syntax of the current statement was not present. For instance, no parenthesis was found after the **`while`** or **`for`** keyword. This error may also occur if a parenthesized expression is incorrectly specified, causing the compiler to end the expression early.

059 `ERROR: Invalid storage class.`

Explanation

A declaration at file scope specified an invalid storage class, such as **`auto`** or **`register`**. This message may be generated if previous errors have caused the compiler to misinterpret statements within a function as file-scope declarations.

060 `ERROR: Incompatible aggregate types.`

Explanation

Incompatible aggregate types were used in an expression, for example, attempting to assign a structure to a structure of a different type.

061 `ERROR/WARNING: Undefined structure/union tag [symbol]`.

Explanation

This message accompanies use of an undefined structure or union tag, that is, a tag for which no members have yet been defined. This is an error if the context requires the size of the type, for instance, if an object of the type is declared or referenced. Otherwise, this is a warning. This warning is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

062 `ERROR: Structure/union type mismatch.`

Explanation

A name declared as a structure tag was used as a union tag or vice versa. Usage of the tag must be consistent within the scope of the tag.

063 `ERROR/WARNING: Duplicate declaration of item [symbol].`

Explanation

The indicated identifier has been defined more than once within the same scope. This error may be generated because of a preceding error, but it is generally the result of improper declarations. This message is issued as a warning if multiple **`__actual`** declarations of an **`__inline`** function are encountered.

064 `ERROR: Structure contains no members.`

Explanation

A declaration of the members of a structure or union did not contain at least one member name.

065 `ERROR: Invalid function definition.`

Explanation

An attempt was made to define a function body within another function or in some other inappropriate context. This message may be produced as the result of a previous error.

066 `WARNING: Invalid array limit expression.`

Explanation

The expression defining the size of a subscript in an array declaration did not evaluate to a nonnegative integer constant. This message also occurs if a 0 length is specified for an inner (that is, not the leftmost) subscript.

067 `ERROR: Illegal object.`

Explanation

A declaration specified an illegal object as defined by the ANSI/ISO Standard. Illegal objects include functions that return arrays, arrays of void, and arrays of functions.

068 `ERROR: Illegal object for structure.`

Explanation

A structure or union declaration included an object declared as a function. Although a structure or union may contain an element of function pointer type, functions are not permitted.

069 `ERROR: Structure includes instance of self.`

Explanation

The structure or union whose declaration was just processed contains an instance of itself, which is illegal. This error may be generated if the asterisk (`*`) is forgotten on a structure pointer declaration or if (due to some intertwining of structure definitions) the structure actually contains an instance of itself.

071 `ERROR: Formal declaration error [symbol]`.

Explanation

A variable was declared before the opening brace of a function, but it did not appear in the list of formal names enclosed in parentheses following the function name.

072 **`ERROR/WARNING: Conflict with prior declaration of external: [symbol].`**

Explanation

An external item has been declared with attributes that conflict with a previous declaration. This error may occur when a function used earlier as an implicit **`int`** function returns some other kind of value. Functions that return a value other than **`int`** must be declared before they are used so that the compiler is aware of the type of the function value.

This message is issued as a warning if the type mismatch is harmless, for example, **`int/long`**, **`const/volatile`**, or **`char/unsigned char`**. The message is issued as an error if the mismatch is harmful.

073 **`ERROR: Declaration expected.`**

Explanation

In processing the declarations of objects, the compiler expected to find another declaration but did not. This error may be caused by a preceding error.

074 **`WARNING: Initializer data truncated.`**

Explanation

An array of **`char`** or **`wchar_t`** was initialized from a string literal, but the literal contained more characters (not including the ending null character) than the size of the array. The excess characters were discarded.

075 **`ERROR: Invalid sizeof expression.`**

Explanation

An invalid expression was the operand of **`sizeof`**. This message also may be generated if an attempt is made to take the size of a structure via a tag that has not been defined, or if an attempt is made to take the size of a function, bitfield, or incomplete type.

076 **`ERROR: Left brace expected.`**

Explanation

The compiler expected but did not find an opening left brace in the current context. This error may occur if the opening brace is omitted on a list of initializer expressions for an aggregate.

077 **`ERROR: Identifier expected.`**

Explanation

While processing a declaration, the compiler expected to find an identifier that was to be declared. This error may occur if the prefixes to an identifier in a declaration (parentheses and asterisks) are specified improperly, or if a sequence of declarations is listed incorrectly.

078 **`ERROR: Undefined statement label [symbol]`**.

Explanation

A `goto` statement referred to a statement label but the compiler could not find the definition of the label.

079 `WARNING: Duplicate enumeration value.`

Explanation

More than one enumeration constant in an enumeration list was assigned the same numeric value.

080 `ERROR: Invalid bitfield.`

Explanation

This error is generated in the following circumstances:

- the number of bits specified for a bitfield is invalid (negative or greater than the number of bits in an `int`)
- the number of bits is specified via a nonconstant (such as a variable)
- the type of the bitfield is invalid (bitfields must be of integral type).

081 `ERROR: Preprocessor limit exceeded.`

Explanation

This message is issued under one of two conditions: there are more than 65535 tokens on a source line, or a macro macro expansion generated more than 65535 tokens. To avoid this error, simplify the source line or, in the case of a macro expansion, simplify the macro.

082 `ERROR: Maximum object/storage size exceeded.`

Explanation

The size of an object exceeded the maximum legal size for objects in its storage class, or the last object declared caused the total size of declared objects for that storage class to exceed the maximum. See the *SAS/C Compiler and Library User's Guide* for the limits for any particular storage class.

083 `WARNING: Reference beyond object size.`

Explanation

An indirect pointer reference (usually a subscripted expression) used an address greater than the size of the object that was used as a base for the address calculation. This message generally occurs when the program refers to an element beyond the declared end of an array.

084 `ERROR/WARNING: Redefinition of preprocessor symbol [symbol].`

Explanation

A warning is issued if the compiler encountered a **`#define`** statement for an already-defined symbol. The symbol and the new statement specified a different replacement text (that is, it was a nonbenign redefinition).

A warning is issued unless the **`redef`** compiler option (or cross-platform **`-W1,-cr`** option) is specified. In that case, the previous declaration is stacked and reappears when a **`#undef`** statement for the symbol appears. The warning message is suppressed unless the **`enforce`** or **`strict`** option (or cross-platform **`-w~n`** or **`-W1,-ll`** option) is specified.

085 `WARNING: Return value mismatch for function value [symbol].`

Explanation

The expression specifying the return value of a function was not the same type as the function itself. If possible, the value specified is converted to the appropriate type. The warning serves as notification of the conversion. The warning can be avoided by using a cast operator to force the return value to the function type.

086 `WARNING: Formal definitions conflict with type list.`

Explanation

The compiler encountered a function definition, and a prototype was in scope for that function. One or more of the formal parameters in the definition conflict in type with the types specified in the prototype.

087 `WARNING: Argument count incorrect.`

Explanation

Although a prototype was in scope for this function, it specified a different number of arguments from the call. A function call may execute correctly with the wrong number of arguments in this implementation, but it is likely to cause portability problems.

088 `WARNING: Argument type incorrect.`

Explanation

Although a prototype is in scope for this function, the type of the argument does not match the type in the prototype. The argument is converted to the correct type. This message does not appear if only a simple widening conversion, such as **`char`** to **`int`**, is needed.

089 **`WARNING: Constant converted to required type.`**

Explanation

This message is the same as message 088 but is produced for an argument that is a constant; the conversion is done at compile time.

090 **`ERROR: Invalid argument type specifier.`**

Explanation

The type of an argument in a function prototype was unrecognized or invalid in this context (for example, **`void`**).

091 **`ERROR: Illegal void operand.`**

Explanation

A **`void`** operand was found in a context where a value is required.

092 **`WARNING: Statement has no effect.`**

Explanation

A statement has no effect (including side effects), as this example illustrates:

```
x+2;
```

Although this statement is a valid C statement, it may not be what the programmer intended to write.

093 **`WARNING: No reference to identifier [symbol]`**.

Explanation

The **`auto`** variable named in the message was declared but never used. Although this is valid C, it may not be what the programmer intended. Perhaps another variable was used where this one should have been used.

094 **`WARNING: Uninitialized auto variable [symbol]`**.

Explanation

The named **`auto`** variable appeared to be used before it was assigned a value. This message may be generated for correct code, if at execution time non-sequential execution (possibly using **`goto`** or **`for`** statements) would cause the variable to be initialized before any use. Note that if the variable is inspected before it is initialized, its value is undefined.

Although uninitialized variables often, by chance, have the value 0, this is not (for **`auto`** variables) guaranteed by the ANSI/ISO Standard language definition. Changing the program or moving the program to another system can change the value and cause undefined program behavior.

095 `WARNING: Symbol previously mapped.`

Explanation

The symbol being mapped by a **`#pragma map`** statement has already been mapped by a previous **`#pragma map`** statement. The first **`#pragma map`** statement takes precedence.

096 `WARNING: New pragma map symbol longer than 8 characters; truncated.`

Explanation

The new symbol to which the old symbol is mapped cannot be longer than eight characters.

097 `WARNING: Mapped name may conflict with other compiler-generated names.`

Explanation

A new symbol violates the guidelines enumerated in Chapter 3, "Compiler Processing and Code Generation Conventions," of the *SAS/C Compiler and Library User's Guide*, and may conflict with symbols generated by the compiler.

099 `ERROR: Attempt to change a const lvalue.`

Explanation

Identifiers with the type qualifier **`const`** cannot be changed. Either the program uses the identifier incorrectly, or the **`const`** type qualifier should be removed from the declaration.

100 `WARNING: No prototype declared for function [symbol].`

Explanation

This message is issued only if the **`reqproto`** option (or cross-platform **`-W1,-cf`** option) is in effect. No prototype is in scope for the function at the time that a call is made to the function.

101 `ERROR: Redundant keywords in declaration.`

Explanation

A declaration contained one or more unnecessary keywords. For example, identifier **`xyz`** is declared as **`long long xyz`**.

102 `ERROR: Conflicting keywords in declaration.`

Explanation

A declaration contained keywords that specified conflicting type specifiers or storage-class specifiers. For example, the declaration **`short long xyz`** contains conflicting type specifiers. The declaration **`extern static xyz`** contains conflicting storage-class specifiers.

103 `WARNING: Uninitialized constant [symbol].`

Explanation

An **auto** or **register const** variable was not initialized. Constant values should be initialized at compile time since they cannot be changed at execution time, and their values will be unpredictable if they are not initialized.

104 **WARNING: Conversion from pointer to const/volatile to pointer to non-const/volatile.**

Explanation

An attempt was made to convert a pointer to a **const** object to a pointer to a non-**const** object, or from a pointer to a **volatile** object to a pointer to a non-**volatile** object. If a pointer to non-**const** is used to modify an object declared as **const**, a protection exception or other incorrect results may occur. If a pointer to non-**volatile** is used to access an object declared as **volatile**, the results are unpredictable.

105 **WARNING: Function prototype not allowed for non-C function (except Assembler).**

Explanation

A prototype cannot be used to declare a function that is declared with one of the interlanguage communication keywords such as **__fortran**, **__cobol**, etc.

107 **ERROR: Too many initializers.**

Explanation

More initializers were specified than the number of aggregate elements to be initialized.

109 **ERROR: Invalid use of type name or keyword.**

Explanation

The user attempted to use a **typedef** name or keyword where an identifier or lvalue was expected.

110 **ERROR: Execution terminated.**

Explanation

End of file (EOF) was found unexpectedly on the compiler input file, for example, in the middle of a function. This error may be caused by mismatched braces, an unclosed comment, or unclosed quotes in a **char** literal.

111 **`WARNING: External name longer than 8 characters [symbol].`**

Explanation

Due to restrictions imposed by the IBM linkage editor, externally visible names are truncated to eight characters in the object deck. If you prefer to use long external names, the first eight characters must be unique or multiple names will resolve to the same linkage-editor symbol. This problem will not occur if the **`extname`** option is specified.

112 **`WARNING: Extraneous information in include file name: [symbol].`**

Explanation

The name of a system header file (that is, one included in angle brackets) ended with a qualifier other than ".h" or ".H". The qualifier was ignored.

```
#include [filename] [name.ext].
```

113 **`WARNING: Unable to open specified file, will attempt to include [filename].`**

Explanation

Mainframe only: The compiler could not open an include file specified as **`#include ''[name.h]''`**. It attempts to recover by opening the file as if it had been specified by **`#include <name.h>`**.

114 **`ERROR: Floating point constant out of machine range.`**

Explanation

A floating-point constant was outside the range of numbers supported by IBM 370 architecture machines, which cannot handle a number whose absolute value is greater than approximately 7.23e75 or a nonzero number whose magnitude is less than approximately 5.4e − 79.

115 **`ERROR: Invalid record size (>1024 bytes).`**

Explanation

Mainframe only: Source fed to the compiler had a line with length greater than 1024. Input to the compiler can be in either fixed or variable length records up to 1024 bytes in length. This message is also produced if an input record contains a null (0x00) character.

116 **`ERROR/WARNING: Undefined enum tag [symbol].`**

Explanation

The indicated **`enum`** tag was not previously defined. The message is issued as an error if an attempt is made to define an enumeration by means of the undefined tag. Declarations that are not definitions cause the message to be issued as a warning.

117 `ERROR: Enum contains no members.`

Explanation

A declaration of an **enum** type did not define at least one enumeration constant.

118 `ERROR: Conflicting use of enum/struct/union tag [symbol].`

Explanation

A name was declared as one sort of tag (**struct**, **union** or **enum**) within a scope, and then referenced as another sort. Usage of the tag must be consistent within its scope.

119 `ERROR: Identifiers missing from definition of function[symbol].`

Explanation

A function definition was given in prototype format, but one or more parameter names were omitted from the prototype. Make sure that a name is specified for each parameter.

121 `WARNING: Hex/octal constant too large (high bits are lost).`

Explanation

A hexadecimal or octal constant specified a value that is too large (greater than 255) to fit into a **char** object. The compiler ignores all but the last eight bits in the constant. For example, if the constant **x1234** is specified, the warning is issued and the constant is treated as equivalent to **x34**. The prefix **L** may be employed to specify a wide character constant as in **L'\x12\x34'**.

122 `WARNING: Missing ellipsis.`

Explanation

The last parameter in a function prototype was followed by a comma. Either the comma is superfluous or an ellipsis (**. . .**), which indicates that unspecified parameters may follow the last declared parameter, is missing.

125 `ERROR: Invalid number.`

Explanation

A token appeared to be a number but contains nonnumeric characters. If the number is intended to be a hexadecimal constant, then one or more of the digits is not in the range 0-f. If it is intended to be a floating-point constant, it does not follow the allowed syntax.

This message can also be generated if a number beginning with a 0 contains an invalid octal digit, for example **08**.

126 **WARNING: #endif, #else, or #elif out of order.**

Explanation

One of the preprocessor directives named was encountered unexpectedly. For instance, a **#endif** directive was read with no previous unmatched **#if** or **#ifdef**.

127 **ERROR: Operand to # operator must be a macro argument.**

Explanation

The operand of a # operator must be one of the arguments calling to the macro.

128 **ERROR: [*text*]**

Explanation

A **#error** directive was encountered by the preprocessor. The text of the message is specified by the **#error** directive.

129 **ERROR: Ambiguous member [symbol].**

Explanation

One of the members declared for an anonymous union has the same name as a member in the containing structure or the same name as a member of another anonymous union in the containing structure. Member names in anonymous unions are in the same name space as member names in the containing structure.

131 **ERROR: Invalid or unsupported use of at-sign operator.**

Explanation

The **@** operator only can be used in the argument list of a function call.

132 **WARNING: Extra tokens after valid preprocessor directive.**

Explanation

One or more unexpected or incorrect tokens were found following a preprocessor directive. The tokens may need to be removed, or they may need to be specified correctly.

133 **ERROR: Cannot redefine macro [symbol].**

Explanation

The **__LINE__**, **__FILE__**, **__STDC__**, **__DATE__**, and **__TIME__** macro names cannot be used in a **#define** or **#undef** preprocessor directive.

134 `ERROR: Too many arguments`

Explanation

The maximum number of macro arguments is 31.

136 `ERROR: Invalid use of register keyword.`

Explanation

The **`register`** storage class specifier can be used only in the declaration of a formal parameter or **`auto`** variable for integer, floating-point, and pointer types.

139 `WARNING: Missing #endif.`

Explanation

The preprocessor expected a **`#endif`** preprocessor directive to terminate a previous **`#if`** or **`#ifdef`** directive.

140 `WARNING: Sizeof operator used on array that has been converted to pointer.`

Explanation

Formal arrays are converted to pointers. When the **`sizeof`** operator is applied to a formal array, the return value is the size of the pointer, not the size of the array.

142 `ERROR/WARNING: Array size never given for [symbol]`.

Explanation

This message occurs for an incomplete array definition. When the size of an array is not defined at the first opportunity, some compilers conforming to ANSI may not accept this even if the size is supplied later. For both externally visible and static arrays, the size must be supplied before the end of the compilation.

143 `ERROR: Object has no address`.

Explanation

`register` variables and bitfields cannot have their addresses taken.

144 `ERROR: Combined storage for strings and constants exceeds maximum.`

Explanation

The combined size of all the character string literals and **`const-`** qualified objects exceeds 2 gigabytes.

145 **WARNING: #include file name or suffix truncated to 8 characters.**

Explanation

Mainframe only: Either the member name or the DDname specified in a **#include** preprocessing directive exceeded eight characters in length. The name is truncated on the right.

147 **WARNING: Conversion between function and data pointers.**

Explanation

The ANSI/ISO Standard states that function pointers cannot be converted to data pointers and vice versa. Even though the conversion can be performed in this implementation, other compilers may not accept the conversion. The compiler flags the statement as a possible portability problem.

148 **WARNING: Use of incomplete struct/union/enum tag [symbol]**.

Explanation

Compilers predating the ANSI/ISO standard may not support this use of an incomplete structure, union or enum type. The compiler flags this statement as a possible portability problem. This message is suppressed unless the **strict** option is used.

149 **WARNING: Undefined struct/union/enum tag in prototype scope [symbol]**.

Explanation

The compiler has detected the use of a tag in a prototype that has not been defined previously. The ANSI/ISO Standard defines a separate scope for tags in a function prototype. Tags defined after the prototype are not in the scope of the prototype and are considered to be in a different scope. **struct**, **union**, and **enum** tags that are used in a function prototype should be defined before the prototype.

150 **ERROR: Conversion to different struct | union | enum type.**

Explanation

An attempt was made to assign the value of a structure, union, or enumeration to one of a different type.

152 **ERROR: Cannot define function via typedef name.**

Explanation

The ANSI/ISO Standard does not allow a function to be defined using a typedef name. Specifically, the following is not permitted:

```
typedef int FNC(void);

FNC f {/* ... */}
```

154 **`WARNING: No prototype declared for function pointer.`**

Explanation

A function call via a function pointer was detected, but there is no prototype in scope for the function pointer. This message is suppressed unless the **`reqproto`** option (or cross-platform **`-W1,-cf`** option) is used.

155 **`WARNING: No statement after label.`**

Explanation

A statement label appeared followed immediately by a closing brace. This syntax is not permitted by the ANSI/ISO standard. You can correct this problem by adding a null statement (**`;`**) after the label.

```
switch(i) {
      .
      .
      .
      default: ;
}
```

156 **`WARNING: Operation/comparison of pointer to int and pointer to [type].`**

Explanation

In many implementations of C, values of type pointer to **`int`** may not be compared with or assigned to variables of type pointer to **`short`** or pointer to **`long`**. The compiler flags the statement as a possible portability problem. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

158 **`ERROR: Invalid type name [symbol].`**

Explanation

This message is issued in two circumstances: first, when a cast contains a name that is not a type name; second, when the **`offsetof`** macro is used incorrectly. This macro requires a structure type name as its first argument.

159 **`WARNING: Use of unary minus on unsigned value [symbol].`**

Explanation

Using a unary minus operator on a variable of unsigned type frequently produces unexpected results because **`unsigned`** types cannot have a negative value. Examine the program to ensure that the operator will have its intended effect. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

161 **`WARNING: No prototype declared at definition for function [symbol].`**

Explanation

A function has been defined for which no prototype is in scope. This message is suppressed unless the **`reqproto`** option (or cross-platform **`-W1,-cf`** option) is used.

162 **`WARNING: Non-ANSI use of ellipsis punctuator.`**

Explanation

The compiler detected an ellipsis (**`...`**) used in a way other than as defined by the ANSI/ISO Standard. The Standard allows an ellipsis only in a function declaration to indicate that the function accepts a variable argument list. It also requires that a prototype contain at least one parameter, and a function may take one or more arguments but not zero arguments.

This compiler supports the following non-standard construction, which indicates that **`func`** is a function returning an **`int`** that takes zero or more arguments:

```
int func(...)
```

However, in such cases, this warning is issued to point out the divergence from standard use.

165 **`WARNING: Use of narrow type in prototype.`**

Explanation

A narrow argument type such as **`char`**, **`short`**, or **`float`** was specified in a prototype. Such arguments will not be widened in a function call. Because the behavior will be different in a context where no prototype is in scope, this message indicates a potential problem if all uses of the function are not within scope of a prototype.

166 **`ERROR: Unrecoverable error or too many errors — terminating the compilation.`**

Explanation

The compiler has detected an error that prevents further compilation, or too many errors have been detected to allow compilation to proceed. Correct the previous error or errors and recompile.

167 **`ERROR: This function type should not be defined in C: [type].`**

Explanation

The **`__asm`** function prefix may only appear in function declarations which are not definitions.

168 **`WARNING: Assignment of long to [symbol].`**

Explanation

This message (which is suppressed by default) is issued if a **`long`** is assigned to a **`short`** or **`int`**. It is intended to warn of possible truncation and portability problems.

169 **`WARNING: Incompatible operands of conditional operator.`**

Explanation

The types of the second and third operands of a conditional operator (**`?`**) do not agree. Use casts to ensure that the types match.

170 **`WARNING: Overflow during operation on constants.`**

Explanation

The compiler detected an overflow condition while evaluating a constant expression. The value of the expression is unpredictable. Correct the expression and recompile.

175 **`ERROR: Structure qualifiers do not match previous declarations of structure [symbol].`**

Explanation

`__alignmem` or **`__noalignmem`** keywords in the named structure declaration did not match a previous declaration of the same tag.

176 **`WARNING: Implicitly promoted formal conflicts with prototype.`**

Explanation

The prototype for a function specified an argument with a narrow type (**`char`**, **`short`** or **`float`**). The function definition was not in prototype format, which implies that all arguments should be widened. This discrepancy between the prototype and the function definition is not standard-conforming, and may lead to incorrect results with some C compilers. Either correct the prototype to specify the widened type, or use the prototype format in the function definition.

180 **`WARNING: No space between macro name and its replacement list.`**

Explanation

The preprocessor detected the end of a macro name but found no white space following the name. This message may occur when an illegal character, such as a dollar sign, is used in a macro name and the **`dollars`** option (or cross-platform **`-Wl,-cd`** option) has not been specified. However, if the only error in the macro was the omission of white space following the macro name, the macro will work as intended.

181 `WARNING: Function declared static but defined external [symbol]`.

Explanation

The declaration of the named function indicated the static storage class, but the function is defined without the static keyword.

182 `WARNING: Static function declared but not defined [symbol]`.

Explanation

The named static function has a declaration but is not defined in the compilation.

183 `WARNING: Inline function declared but not defined [symbol]`.

Explanation

The named in-line function has a declaration but is not defined in the compilation.

184 `WARNING: Multiple characters in integral character constant.`

Explanation

An integral character constant contains more than one single- or multibyte-character constant. This is permitted, but the effect is highly non-portable.

185 `ERROR: Comma expected.`

Explanation

In a function prototype or parameter definition, invalid text was encountered instead of a comma following an argument.

186 `WARNING: Implicit conversion between pointer and scalar.`

Explanation

This message is generated if a **`static`** or **`extern`** pointer variable is initialized using a scalar value. A cast should be used to perform the conversion to a pointer type explicitly.

187 `WARNING: Negative value assigned to unsigned type.`

Explanation

Since an unsigned variable cannot hold negative values, the assignment may not produce the expected results. The message can be eliminated by using a cast, which also clarifies the purpose of the assignment. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

188 **`ERROR: Ambiguous use of ILC language keyword.`**

Explanation

In a complicated declaration, the compiler was unable to determine the function type qualified by a language keyword such as **`__fortran`**. The following code shows a pointer to a function returning a pointer to a function that returns a pointer to **`int`**:

```
__pli int * (* (* ab)( ) ) ( );
```

In this case, however, the compiler cannot tell whether the first function or the second is in PL/I. Specify language keywords directly before the **`*`** (for function pointers) or the name (for functions). For example:

```
int * (__pli * (* ad ) ( ) ) ( );
```

This code shows a pointer to a function returning a pointer to a PL/I function that returns a pointer to **`int`**.

192 **`ERROR: Malformed double-byte characters.`**

Explanation

The compiler used a locale that supports DBCS but the source file contained malformed DBCS characters. A source file containing two successive SO characters, for example, would cause this error.

193 **`WARNING: Mixed types in string concatenation - using original type.`**

Explanation

The compiler attempted to concatenate a character-string literal and a wide-string literal. All string literals in a concatenation must be of the same type. The compiler uses the type of the first string literal in the concatenation.

194 **`ERROR: DBCS used outside of comment, character constant or string.`**

Explanation

The compiler used a locale that supports DBCS, but DBCS characters were found outside of the allowed contexts. The ANSI/ISO Standard specifies that DBCS characters can appear only in comments, characters constants, string literals, and header filenames.

195 **`WARNING: Multiple characters in wide character constant.`**

Explanation

A wide-character constant contained more than one character. Only one wide character will fit into a **`wchar_t`**.

196 `ERROR: AT option required when using @ operator.`

Explanation

The compiler does not accept the `@` operator unless the **`at`** option (or cross-platform **`-Kat`** option) is used. Recompile the program using the **`at`** option (or cross-platform **`-Wl,-ca`** option).

---

198 `WARNING: No external functions or data defined.`

Explanation

This warning is produced at the end of a compilation if no externally visible data or functions are defined in the source module.

---

201 `ERROR: Unable to open pass 2 print file. Using stderr.`

Explanation

Mainframe only: The listing file for phase 2 could not be opened. Phase 2 messages are written to **`stderr`** instead. **`stderr`** also should be inspected for information on why the phase 2 listing file failed to open.

---

202 `WARNING: Empty object module.`

Explanation

The input source file contained no externally visible functions or data so an empty object module was generated. This may be due to conditional compilation or optimization eliminating static functions that could not be called from this or any other compilation.

---

204 `WARNING: Item is wrong size for conversion to pointer.`

Explanation

A **`char`** or **`short`** item was converted to a pointer. However, since all pointers on the IBM 370 are 4 bytes in length, it is unlikely that the input item will convert to a valid pointer. However, code to do the conversion is generated.

---

207 `WARNING: Default section name [first function name] is too long. Section name truncated to 7 characters.`

Explanation

The default section name (the name of the first function in the compilation) was longer than seven characters. The section name is used to create various external names by suffixing it with special characters that cannot be used in valid C names. A section name longer than seven characters is truncated to seven characters before suffixing. Truncation can lead to duplicate names (and an unexecutable program) if another compilation has used the same section name.

Since this message appears for the default section name only, you can avoid it by supplying the **`sname`** option (or cross-platform **`-Ksname`** option), which is limited to seven characters.

---

210 `ERROR: Wrong number of parameters for built-in function. An actual function call has been generated.`

Explanation

Built-in function arguments were incorrect in type or number for the generation of in-line code. Consequently, a true function call was generated instead, which executes more slowly than the in-line code. Furthermore, the arguments may also be incorrect for the true function, causing an error when the function call is executed.

This message is rare because the library prototypes for built-in functions ensure correct parameters in most cases. However, it can occur if a library prototype is removed from the header file.

211 `ERROR: A constant has an invalid type or value, (unsigned) 1 was used instead.`

Explanation

An incorrect type, such as **`double`** instead of **`int`**, was specified for a constant argument to a built-in function. The compiler substitutes an **`unsigned`** constant 1.

212 `ERROR: Maximum number of extended external names exceeded.`

Explanation

The source file declared more extended external names than the compiler could support. This rare occurrence should be reported to the SAS Software Representative for C Compiler Products at your site. Refer to the *SAS/C Compiler and Library User's Guide* for more information about the maximum number of extended external names.

213 `ERROR: Maximum number of extended function names exceeded.`

Explanation

The source file declared more extended function names than the compiler could support. This rare occurrence should be reported to the SAS Software Representative for C Compiler Products at your site. Refer to the *SAS/C Compiler and Library User's Guide* for more information about the maximum number of extended function names.

214 `NOTE: No extended names found.`

Explanation

This message appears if the **`extname`** option (or cross-platform **`-Kextname`** option) is specified and no extended names are found.

215 `NOTE: No extended names defined.`

Explanation

This message appears if the **`enxref`** option (or any of the cross-platform **`-Wl,-xxe`**, **`-Wl,-xxs`**, **`-Wl,-xxx`**, or **`-Wl,-xxy`** options) is specified and no functions or variables with extended names are defined.

216 `ERROR: Unsupported use of __remote with pflocal or __asm function.`

Explanation

An attempt was made to convert a `__local` or `__asm` function pointer to a `__remote function` pointer. This conversion cannot be performed, because the source function pointer contains no PRV information. Note that this message can be generated for an unqualified function pointer if the `pflocal` compiler option is used.

217 `ERROR: Pointer to FILE required for this built-in function parameter.`

Explanation

The first parameter of `__builtin_getc` or the second parameter or `__builtin_putc` must be a pointer to the `FILE` type.

220 `ERROR: Register mask is not a compile-time constant.`

Explanation

The register mask argument to the `_ldregs`, `_stregs`, `_code` , and `branch` functions must be a compile-time constant so the compiler can know what registers are needed.

221 `ERROR: Undefined bits set in register mask.`

Explanation

The register mask had an undefined bit set to 1. Bits 0 through 15 in the register mask represent general purpose registers; bits 16, 18, 20, and 22 represent floating-point registers. The remaining bits must be set to 0. Use combinations of the macro symbols defined in `<regs.h>` to create the register mask.

222 `ERROR: Register mask does not match number of function parameters.`

Explanation

The register mask specified a number of registers that do not match the number of parameters to the function. The register mask used in the `_ldregs` function must define a number of registers that matches the number of parameters (excluding the mask) in the function call.

The register mask used in the `_stregs` function must define a number of registers that either equals the number of other parameters in the function call or exceeds that number by one.

223 `ERROR: Register mask specifies unsupported register(s).`

Explanation

The register mask referred to a register that could not be used. The mask may contain a bit that does not correspond to a hardware register, or may specify a register reserved to the compiler. See the *SAS/C Compiler and Library User's Guide* for information on registers that can be used by machine code functions.

224 **`ERROR: Parameter must be address or pointer.`**

Explanation

The compiler expected a pointer argument in a call to the **`_stregs`** built-in function, but one was not supplied. All arguments to the **`_stregs`** function (except the register mask) must be pointers.

225 **`ERROR: SVC number must be a compile-time constant in the range 0-255.`**

Explanation

The operand of the IBM 370 SVC instruction must have a value between 0 and 255.

226 **`ERROR: DIAGNOSE number unsupported or not a compile-time constant.`**

Explanation

The number specified as the argument to the **`_diag`** function was not a compile-time constant. The number must be a constant, since it forms a part of the generated machine instruction.

227 **`ERROR: Parameter must be a constant in the range 0-0xFFFF.`**

Explanation

Arguments two through five to the **`_code`** function must be halfword constants.

228 **`ERROR: Structure or union parameter cannot be loaded into a register.`**

Explanation

A variable of structure or union type was passed to the **`_ldregs`** function. Inspect the function arguments to see if an **`&`** operator has been omitted.

229 **`ERROR: Inline machine code sequence does not begin with _ldregs`**.

Explanation

A call to **`builtin_code`** (or some other machine code function) was made without a previous call to **`__builtin_ldregs`**, which must immediately precede any sequence of calls to other inline machine code functions. This message can occur if some C code intervenes between a call to **`__builtin_ldregs`** and a call to another inline machine code function.

230 **`ERROR: Register mask does not specify a result register.`**

Explanation

A call was made to **`__builtin_stregs`** with a parameter, but the value of that parameter, the register mask, did not specify a result register.

231 **`ERROR: _stregs not preceded by _ldregs. (_stregs operand may be too complex.)`**

Explanation

Similar to message 229, a call to **`__builtin_stregs`** must be preceded by a call to **`__builtin_ldregs`**. No C code other than calls to inline machine code functions may appear between the call to **`__builtin_ldregs`** and the call to **`__builtin__stregs`**. The generation of code will be halted if the operands of the **`__builtin_stregs`** function are overly complex, since they may force interruption of the inline code sequence to compute the target addresses.

232 **`ERROR: Unsupported opcode, mask, or index register for branch.`**

Explanation

A call to the **`__builtin_branch`** function specified an invalid opcode, register mask, or index register.

236 **`ERROR: Error in debugging file: [description]`**

Explanation

This error will be generated when the compiler encounters problems in either reading or generating the debugging information. On MVS, this is frequently caused by a lack of available space in the PDS where the debugging file is to be stored. The <description> offers some explanation about the problem the compiler encountered.

238 **`ERROR: Problem processing __builtin function '[function name]', possibly incorrect type or number of arguments.`**

Explanation

Either the wrong number or wrong types of parameters was passed to a **`__builtin`** function, causing the function to be replaced with an actual function call. However, the calling function does not have a DSA, and thus cannot support the call.

301 **`WARNING: Indirect reference through NULL pointer.`**

Explanation

The program being optimized contained code to dereference a pointer with a NULL value. The results of such a dereference are undefined but may well include an addressing exception or 0C4 abend.

302 **`WARNING: Type punning involves representation change [symbol]`**.

Explanation

An object defined with one type was used as a dissimilar type, for example, a **`long`** object was accessed as a pointer. This message can occur when an assignment is made to a member of a union, and then the value is accessed from a different member of the union. It can also occur when an address to data of one type is used to access data of a dissimilar type by means of a cast.

303 `WARNING: Reference has overlapping definition [symbol]`.

Explanation

A storage reference was determined by the optimizer to overlap an object or an element of an object. The effect of such accesses is not defined by the ISO/ANSI C standard. The warning indicates that optimization of the code may change its effect from what was intended. If the code cannot be rewritten in a more portable fashion, it is recommended to disable optimization for this source file.

304 `WARNING: Dead assignment eliminated [symbol]`.

Explanation

[symbol] was assigned a value on the indicated line, but the value was never used. The assignment was therefore eliminated.

305 `WARNING: Uninitialized variable [symbol]`.

Explanation

The named identifier was not initialized before it was used. The value resulting is unpredictable.

306 `NOTE: [reason] function inlined: [function name] from line [number].`

Explanation

This message occurs as a consequence of multiple levels of inlining. The value [number] is the line number of the call. The named function was expanded inline on the line cited in the message. The line [number] is included only when the location at which the call was expanded differs from the location of the call.

The [reason] indicates whether the **`__inline`** keyword, **`complexity=[c]`** option, or **`local`** option caused the inlining to occur. The [c] represents the complexity of the function, not the value of the **`complexity=[c]`** option.

Although the cross-platform form of the **`complexity=[c]`** option is **`-Oic=`**[number], the value for [reason] will assume the "complexity" form even when the cross-platform compiler is used.

307 `WARNING: Return value missing in inline function.`

Explanation

The indicated line is a return from a function which has been inlined. A caller of the function uses the returned value, but the **`return`** statement does not specify one.

308 `WARNING: Inline function does not use formal parameter [symbol]`.

Explanation

The indicated line is a call that has been inlined. [symbol] is the name of a formal parameter that is not used in the function.

309 `WARNING: Integer operation overflows.`

Explanation

This message is generated when the cross-platform compiler detects that an integer computation overflows.

315 `WARNING: Static variable is unreachable`

Explanation

Control flow cannot reach a use of this variable. The optimizer removes the variable because the variable is effectively unused, so no storage is allocated for it.

316 `WARNING: Static function is unreachable`

Explanation

This static function cannot be called because control flow cannot reach a call to this function. In some cases this message might be issued because the C++ translator creates static virtual functions which are never actually called, and so can be eliminated by the optimizer.

317 `WARNING: Possibly uninitialized variable`

Explanation

This message is issued because initialization of a variable took place inside conditionally executed code. The optimizer cannot determine whether it is possible for the variable to be referenced before it is initialized. The code may be correct even though the optimizer cannot guarantee this.

400 `WARNING: Invalid comparison of pointer and scalar.`

Explanation

The expression contains a relational or equality operator with a pointer and an arithmetic operand. Either both operands must have arithmetic type or both must be pointers.

401 `ERROR: Illegal pointer subtraction for void pointers.`

Explanation

`void *` pointers cannot be used in arithmetic expressions because they point to objects that have no size.

402 `WARNING: Function declared/defined static was previously declared extern:[symbol].`

Explanation

A declaration or a definition of a static function was found, but a previous declaration of the function omitted the **`static`** keyword. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is specified.

404 `WARNING: Variable name conflicts with typedef name.`

Explanation

An identifier was declared in an inner scope with the same **`typedef`** name that had been declared in an outer scope. The **`typedef`** was hidden in the inner scope declaration. This message is suppressed unless the **`strict`** option is specified.

405 `WARNING: Extraneous braces in initializer expression.`

Explanation

An initializer expression contains more braces than required. Check the syntax of the expression to make sure that it is correct. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is specified.

406 `WARNING: 'If' condition always [true | false].`

Explanation

The compiler determined that the condition in the **`if`** statement was a constant expression. If the condition is true, the compiler only generates code for the **`then`** branch. If the condition is false, the compiler only generates code for the **`else`** branch.

407 `WARNING: Assignment statement as 'if' condition.`

Explanation

An assignment operator (**`=`**) was used when an equality operator (**`==`**) may have been intended.

409 `WARNING: Function declared static within a block.`

Explanation

The compiler ignores the **`static`** keyword if it is used in a function declaration in an inner block. Static functions cannot be declared in inner blocks.

412 `ERROR: Size of object is unknown.`

Explanation

The type of one of the operands in an aggregate (structure or union) assignment is incomplete, and its size cannot be determined. This situation can occur when a structure or union tag declares a variable, but the tag itself has not been defined.

413 **WARNING: Comparison of unsigned value and constant is always [true | false].**

Explanation

Comparison of an unsigned value and a signed constant yielded either a true or false value.

For example, this comparison will always be false because **ui** can never be a negative value:

```
unsigned int ui;
if (ui < 0 ) ...
```

The compiler does not generate code to perform the test, nor does it generate any code for the true branch of the **if** statement.

---

414 **WARNING: Conversion of negative constant to unsigned type.**

Explanation

A negative constant value was assigned to an unsigned variable. The assignment may occur in a function call or an initialization. For example, the following statement sets all bits to 1 in an **unsigned long**:

```
static unsigned long u1 = -1;
```

The compiler issues a warning message in this case. A more correct method of setting all bits to 1 is as follows:

```
static unsigned long u1 = 0xffffffff;
```

---

416 **WARNING: Trailing comma in enum declaration.**

Explanation

An enumerator may be missing in the enumerator list of an **enum** declaration.

---

418 **WARNING: Empty argument to a preprocessor macro.**

Explanation

A null macro argument was specified using two consecutive commas or a comma followed by an open parenthesis. The preprocessor replaced the argument with a string of zero length. For example, in the following macro definition **cat(X,,Z)** produces **XZ**:

```
#define cat(a,b,c) a##b##c
```

This message is suppressed unless the **strict** option (or cross-platform **-W1,-ll** option) is specified.

---

419 `ERROR: Unexpected I/O error.`

Explanation

An I/O error occurred while reading an input source file. Refer to the library messages in **`stderr`** for more information.

420 `WARNING: Argument type is unsupported for __ref functions.`

Explanation

An argument to a **`__ref`** function is of a type that cannot have the **`@`** operator applied to it. For instance, an attempt was made to pass a bitfield to a **`__ref`** function.

421 `WARNING: Symbol exceeds maximum length: [symbol]`.

Explanation

The named [symbol] is longer than the maximum allowable number of characters for symbols of this type. By default, the compiler allows 31 characters for symbols with internal linkage. If the **`extname`** option is specified, an identifier can be up to 64K characters long. This message is suppressed unless the **`mention`** option is specified.

422 `WARNING: Unknown option [name].`

Explanation

The named option has been specified in a **`#pragma options`** statement but is not a recognized compiler option. The option may be misspelled, or it may be a command-line-only option.

423 `WARNING: Option does not accept argument [name].`

Explanation

The named option, specified in a **`#pragma options`** statement, does not accept an argument in parentheses.

424 `WARNING: Invalid argument to option [name].`

Explanation

The argument specified for an option in a **`#pragma options`** statement is invalid or out of range.

426 `WARNING: Invalid option in pragma: [name].`

Explanation

During processing of a **`#pragma options`** statement, the compiler found the specified symbol instead of an option name.

427 `WARNING: Option not allowed in options pragma: [name].`

Explanation

The named option can only be used on the command line.

428 `WARNING:Option can not be negated: [name].`

Explanation

Mainframe only: The named option cannot be prefixed with **no**. For example, **nopagesize** is not a valid option.

429 `WARNING: Static prototype in included file.`

Explanation

A static function has been declared in a header file instead of a primary source file. This warning is suppressed unless the **mention** option (or cross-platform **+n** option) is specified.

430 `WARNING: Non-static prototype in main source file.`

Explanation

An external function was declared in the primary source file instead of a header file. This warning is suppressed unless the **mention** option (or cross-platform **-W1,+n** option) is specified.

433 `WARNING: Only first dimension may be undefined.`

Explanation

Only the first or outer most dimension of a declaration of an array may be left undefined.

435 `WARNING: Assignment to non-const __norent variable will prevent reentrancy.`

Explanation

A variable has been declared as **__norent** in a compilation using the **rent** or **rentext** option, but the **const** qualifier was not specified. This implies that the variable may be assigned elsewhere in the program. If, in fact, the variable is assigned elsewhere, the program will be non-reentrant.

436 `WARNING: Initialization prevents reentrancy.`

Explanation

The initialization of the variable will prevent the program from being reentrant. This message is generated because one or more elements of the item being initialized have a type which cannot be initialized before execution time. See the discussion of the `__norent` keyword in the *SAS/C Compiler and Library User's Guide* for more information

---

437 `WARNING: Operation/comparison of pointer to char and pointer to [type].`

Explanation

In some implementations of C, values of type pointer to `char` may not be compare or assigned to pointers of other types. The compiler flags the statement as a possible portability problem.

This message is suppressed unless the `strict` option (or cross-platform -W1,-ll option) is used.

---

438 `WARNING: Application of this linkage request is invalid.`

Explanation

`#pragma linkage` was specified for an identifier declared with a conflicting keyword, such as `__ref`. Either remove the conflicting keyword or the `#pragma linkage` directive.

---

441 `WARNING: Assignment of int to short.`

Explanation

A `signed` or `unsigned int` was assigned to a `short` or `unsigned short`. Data truncation may occur. This message is suppressed unless the `strict` option (or cross-platform `-W1,-ll` option) is used.

---

442 `WARNING: Function declared with two different types: [function name]`

Explanation

The function specified in the message was declared as having two conflicting types.

---

443 `WARNING: Assignment of short to char.`

Explanation

A `signed` or `unsigned short` was assigned to a `char` (possibly `signed` or `unsigned`). Data truncation may occur. This message is suppressed unless the `strict` option (or cross-platform `-W1,-ll` option) is used.

---

444 `WARNING: Assignment of int to char.`

Explanation

A **`signed`** or **`unsigned int`** was assigned to a **`char`**. Data truncation may occur (possibly **`signed`** or **`unsigned`**). This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

445 `WARNING: Assignment of long to char.`

Explanation

A **`signed`** or **`unsigned long`** was assigned to a **`char`**. Data truncation may occur (possibly **`signed`** or **`unsigned`**). This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

446 `WARNING: Assignment of long to short.`

Explanation

A **`signed`** or **`unsigned long`** was assigned to a **`signed`** or **`unsigned short`**. Data truncation may occur. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

447 `WARNING: Constant assignment too large for char.`

Explanation

A constant with a value greater than the maximum value for type **`char`** was assigned to a variable of type **`char`**. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

448 `WARNING: Constant assignment too small for char.`

Explanation

A constant with a value smaller than the minimum value for type **`char`** was assigned to a variable of type **`char`**. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

449 `WARNING: Constant assignment too large for unsigned char.`

Explanation

A constant with a value greater than 255 was assigned to a variable of type **`unsigned char`**. Data truncation will occur.

450 `WARNING: Constant assignment too small for unsigned char.`

Explanation

A constant with a value less than 0 was assigned to a variable of type **`unsigned char`**. Data truncation will occur.

451 `WARNING: Constant assignment too large for signed char.`

Explanation

A constant with a value greater than 127 was assigned to variable of type **`signed char`**. Data truncation will occur.

452 `WARNING: Constant assignment too small for signed char.`

Explanation

A constant with a value less than -128 was assigned to a variable of type **`signed char`**. Data truncation will occur.

453 `WARNING: Assignment of long constant to char.`

Explanation

A constant of type **`long`** was assigned to a variable of type **`char`**. This may indicate a portability problem, as data truncation could occur on a different platform. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

454 `WARNING: Constant assignment too large for short.`

Explanation

A constant whose value is greater than the maximum value for the type **`short`** was assigned to a variable of type **`short`**. Data truncation will occur.

455 `WARNING: Constant assignment too small for short.`

Explanation

A constant with a value smaller than the minimum value for type **`short`** was assigned to a variable of type **`short`**. Data truncation will occur.

456 `WARNING: Constant assignment too large for unsigned short.`

Explanation

A constant whose value is greater than the maximum value for the type **`unsigned short`** was assigned to a variable of type **`unsigned short`**. Data truncation will occur.

457 `WARNING: Constant assignment too small for unsigned short.`

Explanation

A constant with a value smaller than the minimum value for type **`unsigned short`** was assigned to a variable of type **`unsigned short`**. Data truncation may occur.

459 `WARNING: Assignment of long constant to short.`

Explanation

A constant of type **`long`** was assigned to a variable of type **`short`**. This may indicate a portability problem, as data truncation could occur on a different platform. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

460 `WARNING: Constant assignment too large for int.`

Explanation

An unsigned constant whose value is greater than the maximum **`unsigned`** value for the type **`int`** was assigned to a variable of type **`int`**. Data truncation will occur. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

461 `WARNING: Constant assignment too small for int.`

Explanation

A constant whose value is less than the minimum value for the type **`int`** was assigned to a variable of type **`int`**. Data truncation will occur. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

462 `WARNING: Constant assignment too large for unsigned int.`

Explanation

A constant whose value is greater than the maximum value for the type **`unsigned int`** was assigned to a variable of type **`unsigned int`**. Data truncation will occur. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

463 `WARNING: Constant assignment too small for unsigned int.`

Explanation

A constant with a value less than 0 was assigned to a variable of type **`unsigned int`**. The constant will be converted to a large **`unsigned`** value, which may not be the intended effect.

465 `WARNING: Assignment of long constant to int.`

Explanation

A constant of type **`long`** was assigned to a variable of type **`int`**. The assignment will behave properly on the 370, but may not if ported to another machine architecture. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

469 `WARNING: Constant assignment too small for unsigned long.`

Explanation

A constant with a value less than 0 was assigned to a variable of type **`unsigned long`**. The constant will be converted to a large **`unsigned long`** value, which may not be the intended effect.

470 `WARNING: Assignment of unsigned constant to long.`

Explanation

A constant of type **`unsigned long`** whose value is greater than the maximum value of the type **`long`** was assigned to a variable of type **`long`**. Data truncation will occur. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

473 `WARNING: Operation/comparison of unsigned *char and signed *char.`

Explanation

An assignment was made between **`signed`** and **`unsigned char`** pointers. This message is suppressed unless the **`strict`** option (or cross-platform **`-W1,-ll`** option) is used.

476 `WARNING: Invalid control expression in for statement.`

Explanation

The control (second) expression of a **`for`** statement had **`void`** type. The control expression must either have an arithmetic or pointer type, or be omitted.

477 `WARNING: Unmatched quote in #pragma title preprocessor command.`

Explanation

An unmatched quote was found in processing a **`#pragma`** title compiler directive.

Unnumbered Error and Warning Messages

These messages are compiler diagnostic notes concerning the following:

- option processing
- environment conditions
- the global optimization phase.

Option Processing Messages

These messages generally describe conditions that arise during compiler option processing. The messages are generally for diagnostic purposes only; the compiler ignores the unrecognized string and continues processing. Check the Options in Effect part of the compiler listing to determine the options actually used by the compiler.

Invalid DDname prefix specified
Mainframe only: The argument to the **files** compiler option contains characters that cannot be used in a DDname, or the argument is longer than three characters (MVS, TSO only).

Invalid default bitfield type.
The allocation unit specified with the **bitfield** option must be either 1, 2, or 4. The **bitfield** option (or cross-platform **-Kbitfield** option) is ignored.

Invalid GO complexity() option: 0-20 allowed - overridden.
The **complexity** option (or cross-platform **-Oic=**[number] option) requires an integer value between 0 and 20, inclusive. The compiler will use a default value of 0.

Invalid GO depth() option: 0 -6 allowed - overridden.
The **depth** option (or cross-platform **-Oid=**[number] option) requires an integer value between 0 and 6, inclusive. The compiler will use a default value of 3.

Invalid GO inline option [*text*].
The specified option is an unknown optimization phase option. It is ignored.

Invalid GO rdepth() option: 0-6 allowed - overridden.
The **rdepth** option (or cross-platform **-Oir=**[number] option) requires an integer value between 0 and 6, inclusive. The compiler will use a default value of 1.

Invalid LISTING disposition specified
Mainframe only: An attempt was made to route the compiler listing to a destination other than a minidisk, SFS directory, virtual printer, or the terminal (CMS only).

Invalid multiple specified for zapspace size.
The **zapspace** option (or cross-platform **-Kzapspace** option) requires an integer value between 0 and 22, inclusive. A value was found that did not fit these criteria, so the compiler used the default value of 1.

Invalid number of lines per page specified for listing
The number specified by the **pagesize** option was not greater than 10 and less than 255. The compiler sets the page size to 55.

Unknown option [*xxx*] [*yyy*].
The compiler did not recognize [xxx] as an option and assumed that [yyy] was intended. The message identifies the bad option and the assumed option.

Invalid [*option-type*] option: [*text*]
An invalid option was specified. The message may classify the option by [option-type]. The [text] may identify the string that was unrecognizable or may indicate what action the compiler took (or both).

Invalid section name specified.
The **sname** option (or cross-platform **-ksname** option) specified a string that either contained invalid characters or was longer than seven characters.

`Invalid symbol definition.`
An attempt was made to define an invalid preprocessor symbol with the **`define`** option (or cross-platform **`-Dsym=`**[value] option).

`Invalid value specified for minimum zapspace size.`
The **`zapmin`** option (or cross-platform **`-Kzapmin`** option) requires an integer value between 24 and 512, inclusive. The compiler will use a default value of 1.

Environmental Conditions

The messages in this section describe error conditions in the environment, rather than errors in the source file caused by improper language specifications. In most cases, errors written by the run-time library to the standard error file, **`stderr`**, provide additional information.

`Can't close file for debugger symbol information.`
The second phase of the compiler encountered an error while closing the debugger symbol file.

`Can't create file for debugger symbol information.`
The second phase of the compiler was unable to create the debugger symbol file.

`Can't create object file.`
The second phase of the compiler was unable to create the object file.

`Can't create quad file.`
The first phase of the compiler was unable to create the quad file.

`Can't open quad file.`
The second phase of the compiler was unable to open the quad file.

`Can't open source file.`
The first phase of the compiler was unable to open the source file.

`End-of-file on object file.`
The second phase of the compiler encountered a premature end-of-file on the object file.

`Error closing object file.`
The second phase of the compiler encountered an error while closing the object file.

`Intermediate file error.`
The first phase of the compiler encountered an error when writing to the quad file.

`Invalid quad file.`
The quad file was created incorrectly by the first phase of the compiler or has been corrupted between compiler phases.

`No functions or data defined.`
A source file that did not define any functions or data elements was processed by the compiler. This error always terminates execution of the compiler. It can be generated by forgetting to terminate a comment, which then causes the compiler to treat the entire file as a comment.

`Not enough memory.`
This message is generated when either phase of the compiler uses up all the available working memory. The only remedy for this error is either to increase the available memory or (if the maximum is already available) to reduce the size and complexity of the source file.

`Object file error.`
The second phase of the compiler encountered an error when writing to the object file.

Global Optimization Phase Messages

These messages can be issued by the global optimization phase:

`Can't reopen output quad file.`
The optimizer could not open the quad file. Check for a library diagnostic message containing more information about why the file could not be opened.

`Error while replacing quad file.`
The optimizer could not close the quad file. Check for a library diagnostic message containing more information about why the file could not be closed.

`Quad file not replaced.`
The optimizer could not delete the existing copy of the quad file. This usually is caused by write-protecting the quad file.

3 LSCF Full-Screen Support Library Messages

This chapter presents in table format diagnostics generated by the full-screen support library. Some error and warning codes are used in more than one function; in such cases, possible substitutions are listed in the description.

Note: The LSCF prefix has been omitted from the descriptions below.

Table 3.1 *LSCF Messages*

Message	Return Code	Description	Function
001	– 1	**FS_INITERR** Initialization has already taken place.	**fsinit**
		Initialization has *not* taken place. Use **fsinit** to establish the full-screen environment.	all other functions
002	– 2	**FS_INITFAIL** Initialization aborted; unable to find **L$FSSL** load module.	**fsinit**
003	– 3	**FS_NOMEM** (Note: This can be issued from any routine.) Memory allocation failed; [text] cannot be established.	
		Values of [text]	
		field definition	**fsdffd**
		panel definition	**fsdfpn**
		view definition	**fsdfvw**
		full-screen environment	**fsinit**
		default view	
		modified fields array	**fsrdpn**
004	– 4	**FS_BADFREE** Memory deallocation failed; use **=storage** to locate problem.	**fsenpn** **fsenvw** **fsrdpn** **fsrmfd** **fsterm**
005	– 5	**FS_INVLVIEW** View ID specified is invalid: [text].	
		Values of [text]	
		ID must be positive	**fsdfvw** **fsenvw**
		update or display pending	**fsdspn**
		view ID unknown	**fsuppn**
006	– 6	**FS_DUPLVIEW** Viewing area is already defined. Either specify another view or end current view first.	**fsdfvw**
007	– 7	**FS_USEDVIEW** View currently requested is the one being displayed.	**fsdspn**

Message	Return Code	Description	Function
008	– 8	**FS_OVLPVIEW** Requested viewing area overlaps viewing area for pending output; output pending for view ID=[view ID number].	**fsdspn**
009	– 9	**FS_INVLPAN** Panel has not been defined. Panel has not been defined or panel name is invalid. Panel has not been defined or has not been previously displayed (use **fsdspn** first).	**fsdffd** **fsrmfd** **fsenpn** **fsrdpn** **fsuppn**
010	– 10	**FS_DUPLPAN** Panel is already defined; panel= [panel name]. Either specify another panel ID or end current panel first.	**fsdfpn**
011	– 11	**FS_INVLFLD** Specified field ID is invalid: [text]; field ID=[field ID number]. Values of [text] must be positive has never been defined	 **fsdffd** **fsdspn** **fsrmrd** **fsuppn**
012	– 12	**FS_OVLPFLD** Field area specified overlaps existing field with ID=[field ID number].	**fsdffd**
013	– 13	**FS_DUPLFLD** Field ID specified has already been used.	**fsdffd**
014	– 14	**FS_INVLLEN** Invalid field length: [text]; max length=[length]. Values of [text] / Values of [length] length must be > 1 or length too long / field length or panel length	**fsdffd**
015	– 15	**FS_INVLOFST** Invalid display offset specified; offset must be within range 1 to [field_length].	**fsdspn** **fsuppn**

Message	Return Code	Description		Function
016	– 16	**FS_INVLHGHT** Invalid height specified for [item] definition; must be between 1 and [num].		
		Values of [text]	Values of [num]	
		panel	99999 (infinity)	**fsdfpn**
		view	terminal height	**fsdfvw**
017	– 17	**FS_INVLWIDTH** Invalid width specified for [item] definition; must be between 1 and [max].		
		Values of [item]	Values of [max]	
		panel	99999 (infinity)	**fsdfpn**
		view	view terminal width	**fsdfvw**
018	– 18	**FS_INVLROW** Invalid row for [item]; row must be between 1 and [max].		
		Values of [item]	Values of [max]	
		field definition	panel height	**fsdffd**
		terminal definition	62 (current maximum)	**fsdftm**
		view definition	terminal height	**fsdfvw**
		panel display	panel height	**fsdspn**
		panel update	panel height	**fsuppn**
019	– 19	**FS_INVLCOL** Invalid column for [item]; column must be between 1 and [max].		
		Values of [item]	Values of [max]	
		field definition	panel width	**fsdffd**
		terminal definition	160 (current maximum)	**fsdftm**
		view definition	terminal width	**fsdfvw**
		panel display	panel width	**fsdspn**
		panel update	panel width	**fsuppn**
020	– 20	**FS_INVLCOLR** Requested color invalid for terminal; check local terminal guide for details on colors supported.		**fsdftm**
021	– 21	**FS_INVLNUMCOLR** Invalid number of colors requested; range must be between 0 and 7.		**fsdftm**

Message	Return Code	Description	Function
022	– 22	**FS_OUTPEND** [operation] not allowed while output pending. Values of [operation] field definition panel display ending a panel ending a view field removal panel update	 **fsdffd** **fsdspn** **fsenpn** **fsenvw** **fsrmfd** **fsuppn**
023	– 23	**FS_INPTPEND** [operation] not allowed while input pending. Values of [operation] field definition panel display ending a panel ending a view field removal panel update	 **fsdffd** **fsdspn** **fsenpn** **fsenvw** **fsrmfd** **fsuppn**
024	– 24	**FS_NULLDATA** Invalid **datavalue** parameter; parameter cannot be a **NULL** pointer.	**fsdffd**
025	– 25	**FS_NOTERM** Internal error = **NOTERM**. Contact SAS Technical Support.	
026	– 26	**FS_NOT3270** Internal error = **NOT3270**. Contact SAS Technical Support.	
027	– 27	**FS_INVTBL** Invalid terminal translate table. The value of table type for **fsdfch** must be one of **FS_TR_INB**, **FS_TR_OUB**, **FS_GE_INB**, or **FS_GE_OUB**.	
028	– 28	**FS_OPENFAIL** Internal error = **OPENFAIL**. Contact SAS Technical Support.	
030	– 30	**FS_INVLKEY** PF key value is not valid. **fsispfk** PF key value must be in the range ≥ 1 and ≤ 24.	

Table 3.2 *Warning Messages*

Warning	Return Code	Description	Function
`FS_ADJATTR`	0x00000010	Adjusted the field attributes.	`fsdffd`
`FS_ADJCOL`	0x00000001	Adjusted the viewing area's beginning column value.	`fsdfvw`
`FS_ADJCOLR`	0x00000008	Adjusted the field's color.	`fsdffd`
`FS_ADJHGHT`	0x00000002	Adjusted height of one or more of existing viewing areas.	`fsdftm`
`FS_ADJWIDTH`	0x00000004	Adjusted the viewing area width.	`fsdfvw` `fsdftm`
`FS_DFLTVIEW`	0x00001000	Default viewing area was used.	`fsdspn`
`FS_DUPLCOLR`	0x00000800	Duplicate color values specified.	`fsdftm`
`FS_IGNATTR`	0x00000040	Ignored superfluous attributes.	`fsdffd`
`FS_IGNFLAG`	0x00000080	Ignored superfluous display flags.	`fsdspn`
		Ignored superfluous update flags.	`fsuppn`
`FS_IGNOPTN`	0x00000200	Ignored superfluous options.	`fsdftm`
`FS_IGNSIZE`	0x00000100	Ignored changing the size of the screen currently being used (input or output is pending).	`fsdftm`
`FS_NOATTR`	0x00000020	No field attributes supplied.	`fsdffd`
`FS_NOINPUT`	0x00000400	No input exists.	`fsrdpn` `fsrdsc`

Table 3.3 *General Messages*

Message	Description
098	Cannot load message text (`L$FEMSG`) module.
099	SAS/C FSSL Release [num] (resident), [num] (transient) [os], where [num] are release numbers (like 5.01B) and [os] is an operating system (like MVS/SP or CMS).

4 LSCI ILCLINK Messages

This chapter describes ILCLINK diagnostic messages, their severity levels, causes, and resolutions.

ILCLINK diagnostic messages have the form

```
LSCI[num] [severity]: [message text]
```

where

NOTE	0
WARNING	4
ERROR	8
SEVERE	12
INTERNAL	(ABEND)

A NOTE usually displays information about the expected behavior of the linker. WARNING messages indicate an error has occurred that will not affect ILCLINK's processing but should be examined by the user. An ERROR or SEVERE message is produced if ILCLINK encounters a condition that will not allow processing to continue. An ERROR indicates a problem that is within the user's control, such as an invalid control statement. A SEVERE message indicates a problem with the environment, such as an out-of-memory condition. Finally, an INTERNAL message (of which 038 is the only example) indicates that ILCLINK has failed to process a correct input file. ILCLINK issues a user ABEND in this situation. ILCLINK will continue processing if no diagnostics are produced with a severity level greater than WARNING.

Unless otherwise indicated, these messages can occur in any operating environment.

Note: The ILCLINK message prefix LSCI has been omitted from the descriptions below.

ILCLINK Messages

000 **`SEVERE: Not enough memory to continue.`**

Explanation

An attempt to allocate memory failed.

Action

Under CMS, increase the size of the virtual machine. Under TSO and MVS-batch, increase the region size.

001 `SEVERE: Internal error in [function]. Code [num] .`

Explanation

An internal error was detected in the specified function.

Action

Report the function name and code to the Technical Support Department at SAS Institute.

---

002 `WARNING: Input file name truncated to 8 characters.`

Explanation

CMS only: The filename given as the name of the input file contains more than 8 characters. The name will be truncated to 8 characters.

Action

CMS filenames cannot have more than 8 characters.

---

003 `ERROR: Missing input file name.`

Explanation

CMS only: No input filename was specified on the command line.

Action

Re-invoke ILCLINK with the name of the input file as the first parameter.

---

004 `WARNING: Unknown option [option] ignored.`

Explanation

The specified option is not an option. The option may have been misspelled.

Action

Re-invoke ILCLINK with the correct option name.

---

005 `ERROR: Missing right parenthesis.`

Explanation

TSO, MVS-batch only: An option requiring a parenthesized value does not have the terminating right parenthesis.

Action

Re-invoke ILCLINK with the correct form of the option.

---

006 `ERROR, WARNING: Unable to open [file].`

Explanation

ILCLINK could not open the specified file. The error level is Warning if the file is an output file, and Error if the file is the input file.

Action

Refer to library messages on `stderr` for more information about the error.

---

007 `ERROR: Invalid parameter [parameter] .`

Explanation

CMS only: More than one non-option parameter was used.

Action

Re-invoke ILCLINK using the filename of the input file as the only non-option parameter.

---

008 `SEVERE: Unable to determine operating system.`

Explanation

ILCLINK cannot determine the operating system.

Action

Contact the Technical Support Department at SAS Institute.

---

009 `ERROR: Expecting [statement-types] statement.`

Explanation

The input file contains control statements that are not in the expected order. The message specifies the types of statements that can occur in the current context.

Action

Correct the statements in the input file.

---

010 `ERROR: Incorrect or missing data in statement.`

Explanation

The statement contains values that are misspelled, in the wrong order, or otherwise incorrect, or the statement does not contain an expected value or keyword.

---

011 `ERROR: Only one entry point language may be specified.`

Explanation

More than one language name was found in the FIRST statement.

Action

Correct the FIRST statement so that only one language name is used.

---

012 `ERROR: Reading input file.`

Explanation

An error occurred while reading the input file.

Action

Refer to library messages on **`stderr`** for more information about the error.

---

013 `ERROR: Statement longer than [num] characters.`

Explanation

A statement in the input file is longer than the maximum allowed. This can occur if the input file has a logical record length greater than 255.

Action

Reformat the input file to use only statements less than or equal to 255 characters.

---

014 `ERROR: No PROCESS statements in input file.`

Explanation

No PROCESS statement was found in the input file. A PROCESS statement must be used to produce useful results.

Action

Add PROCESS statements as required to link the program.

---

015 `ERROR: Only one FIRST statement may be used.`

Explanation

More than one FIRST statement was found in the input file. There may be only one FIRST statement.

Action

Remove extra FIRST statements from the input file.

---

016 `ERROR: Writing to [file] .`

Explanation

An error occurred while writing to the specified output file.

Action

Refer to library messages on **`stderr`** for more information about the error.

---

017 `NOTE: No non-C languages defined.`

Explanation

No languages other than C were named in any FIRST or LANGUAGE statement.

Action

None. This message is for information only.

---

018 `ERROR: Unknown PROCESS keyword [keyword] .`

Explanation

The PROCESS statement keyword is not CLINK, LINK, LKED, LOAD, or GENMOD. One of these keywords must be specified in a PROCESS statement.

Action

Correct the PROCESS statement.

---

019 `ERROR: Opening utility file.`

Explanation

An error occurred while opening the utility file.

Action

Refer to library messages on **`stderr`** for more information about the error.

---

020 `NOTE/WARNING/ERROR: Return code from [command] was [num] .`

Explanation

The return code from the specified operating system command or utility is the value shown in the message.

Action

If the return code is unexpected, consult the appropriate operating system documentation for the meaning of the return code. The operating system may have issued other messages, or the library may have issued a diagnostic message on **`stderr`**.

---

021 `WARNING: SYSTEM command not issued due to [reason] .`

Explanation

Either an error or attention prevented the command in the SYSTEM statement from completing.

Action

If an error occurred, refer to the library messages on **`stderr`** for more information.

---

022 `WARNING: Entry point name longer than 8 characters. Using [name] as the entry point name.`

Explanation

The entry point name specified in a FIRST statement, linkage editor ENTRY statement, or the RESET option for the LOAD command is longer than 8 characters.

Action

Shorten the entry point name.

---

023 `NOTE: Languages in program are [language-list] .`

Explanation

The list contains the language names specified in the FIRST and LANGUAGE statements.

Action

None. This message is for information only.

---

024 `WARNING: Name [library] truncated to eight characters.`

Explanation

A library name (a filename under CMS, or a DDname under MVS-batch and TSO) in an AUTOCALL statement is longer than eight characters. The maximum length of these names is eight.

Action

Correct the AUTOCALL statement.

---

025 `ERROR: No more than [num] TXTLIBs may be GLOBALed.`

Explanation

CMS only: Prior to VM/SP Release 5, the maximum number of GLOBALed TXTLIBs is 8. The total number of TXTLIB names found in the AUTOCALL statement exceeds this number.

Action

Remove TXTLIB names from the AUTOCALL statements. If the application requires more than 8 TXTLIBs, you must merge some.

---

026 `NOTE: Echo [command] .`

Explanation

This message is issued when the ECHO option is in effect. ILCLINK has issued the specified operating system command.

Action

None. This message is for information only.

---

027 `ERROR: Unknown CLINK option [option] .`

Explanation

The specified option in a PROCESS CLINK statement is not a CLINK option. The option may have been misspelled.

Action

Refer to CLINK documentation for a list of options.

---

028 `ERROR: Entry point [name1] in [statement1] conflicts with entry point [name2] spec via [statement2] .`

Explanation

Two conflicting entry point names have been detected. The name of the entry point may be specified by a FIRST statement, a linkage editor ENTRY statement, or the RESET option in a PROCESS LOAD statement.

Action

Check the specified statements and ensure that the entry point names do not conflict.

---

029 `WARNING: You must specify an explicit entry point name for [language] programs.`

Explanation

The FIRST statement specified a default entry point name and no default can be chosen. An explicit entry point name must be specified when the FIRST language is COBOL, FORTRAN, or a user-supported language.

Action

Determine the entry point name and add it to the FIRST statement.

---

030 `WARNING: [DDname] FILEDEF is already in effect.`

Explanation

CMS only: A FILEDEF for a TXTLIB named in an AUTOCALL statement has already been issued. ILCLINK will not reissue the FILEDEF.

Action

Clear conflicting FILEDEFs before invoking ILCLINK.

---

031 `WARNING: Concatenated SYSLIB FILEDEFs are not supported by the LKED command.`

Explanation

CMS only: More than one AUTOCALL library name was specified for a PROCESS LKED statement. As of Release 5 of CMS, the LKED command does not support concatentated SYSLIB libraries. ILCLINK will issue the FILEDEFs anyway.

Action

Put all autocalled object code into a single TXTLIB.

---

032 `WARNING: Invalid DDname prefix [prefix] specified by FILES option.`

Explanation

TSO, MVS-batch only: The value specified by the FILES option was greater than 3 characters long, or the first character was not A– Z, #, $, or @.

Action

Specify a valid DDname prefix with the FILES option.

---

033 `ERROR: Dataset is already open.`

Explanation

TSO, MVS-batch only: A data set referred to in the current statement is already open.

Action

Ensure that all data sets that will be used are closed before invoking ILCLINK.

034 `ERROR: Specified DDname not found.`

Explanation

TSO, MVS-batch only: A DDname specified in the current statement has not been allocated.

Action

Ensure that all DDnames that will be used are allocated before invoking ILCLINK.

035 `SEVERE: In dynamic allocation. Return code [num], reason code [num], information reason [num].`

Explanation

TSO, MVS-batch only: An unexpected error occurred while executing SVC 99 (dynamic allocation).

Action

Report the return code, reason code, and information reason code to the Technical Support Department at SAS Institute.

036 `SEVERE: Dynamic allocation request denied by installation. Return code [num], reason code [num], information reason [num].`

Explanation

TSO, MVS-batch only: An SVC 99 request for dynamic allocation failed due to an installation restriction.

Action

Consult a systems programmer at your site for more information.

037 `ERROR: JOBLIB, STEPLIB, JOBCAT, or STEPCAT specified on [statement-type] statement.`

Explanation

TSO, MVS-batch only: One of the above DDnames was used in a statement.

Action

Use another DDname to refer to the data set.

---

038 `INTERNAL: Invalid dynamic allocation parameter list.`

Explanation

TSO, MVS-batch only: ILCLINK created an invalid SVC 99 parameter list. ILCLINK will issue a user ABEND.

Action

Report the error, including the traceback, to the Technical Support Department at SAS Institute.

---

039 `ERROR: Required DDname unavailable.`

Explanation

TSO, MVS-batch only: A DDname in the current statement has not been allocated.

Action

Ensure that the DDnames used in the input file have been allocated before invoking ILCLINK.

---

040 `ERROR: Unable to allocate [dsname] OLD. Allocated to another job or user.`

Explanation

TSO, MVS-batch only: The specified data set is already in use by another job or session.

Action

Free any conflicting allocation(s) before invoking ILCLINK.

---

042 `ERROR: Deconcatenation of DDname [ddname] would result in duplicate DDnames.`

Explanation

TSO, MVS-batch only: A data set concatenation created by ILCLINK contains a DDname that has since been allocated.

Action

Use unique DDnames in ALLOCATE commands issued via the SYSTEM statement.

---

043 `ERROR: SYSTEM statement too long to process.`

Explanation

A command specified in a SYSTEM statement exceeds the limit of 500 characters.

Action

Reduce the number of characters in the command. This error is most likely caused by excessive blank space on the command.

---

044 `WARNING: SYSTEM statements may not be issued in MVS-batch.`

Explanation

MVS-batch only: ILCLINK will not issue a command in a SYSTEM statement under MVS-batch.

Action

Remove the statement.

---

045 `ERROR: GENMOD command too long to issue.`

Explanation

CMS only: A PROCESS GENMOD statement requires that a GENMOD command longer than 240 bytes be created. ILCLINK cannot create commands longer than 240 bytes.

Action

Re-code the PROCESS GENMOD statement. This error is most likely the result of excessive blank space in the statement.

---

046 `ERROR: Cannot determine first CSECT name.`

Explanation

CMS only: An error occurred while reading the LOAD MAP file.

Action

Refer to library messages for more information about the error.

047 `ERROR: [utility] abnormally terminated.`

Explanation

TSO, MVS-batch only: The utility (for example, the linker) invoked for the current process ABENDed.

Action

Refer to diagnostics issued by the utility for more information.

048 `ERROR: Duplicate DDnames specified in concatenation.`

Explanation

A DDname concatenation used the same DDname more than once. ILCLINK cannot deconcatenate these DDnames.

Action

Change the JCL used to run ILCLINK to specify unique DDnames.

5 LSCL CLINK Messages

This chapter documents messages generated by the CLINK utility (either mainframe or cross-platform). Each message contains the message text and an explanation of the message, as well as operating-system information and the action needed to correct the error, if applicable. All messages can occur in any operating environment unless otherwise indicated. In some cases, messages from the run-time library written to the standard error file **stderr** may further describe the problem.

CLINK diagnostic messages have the form

```
LSCL[num] [severity]: [message text]
```

where

num	is the message number
severity	NOTE, WARNING, or ERROR
message text	is a description of an action (NOTE), possible problem (WARNING), or condition that will prevent further processing (ERROR).

Messages are written to **stdout** if the **term(-t)** option is in effect; they are written to the listing if a listing is created. If a WARNING message is generated, the final return code from CLINK will be no less than 4. If an ERROR message is generated, CLINK will terminate immediately with a return code of 8.

Note: The CLINK message prefix LSCL has been omitted from the descriptions below. The prefix LSCL does not appear with the cross-platform CLINK.

LSCL Messages

003 **ERROR: Error occurred while attempting to write CLINK output file.**

Explanation

An attempt to write one or more items to the output file stream has been unsuccessful. Usually this is caused by having insufficient space available for all the output, but any file system problem or failure that might cause **fwrite()** to fail could also cause this message.

Action

Re-run CLINK with the =warning run-time option to provide additional SAS/C Library diagnostics. Make sure the space available for the output file is large enough to hold all the output.

004 `ERROR: Invalid CLINK input [first_10_bytes]`

Explanation

CLINK did not recognize a record of the input as either a control statement or an object code record, and displayed the first 10 bytes of the card.

Action

All input to CLINK must be valid object modules or control statements. This message can be caused by attempting to use a load module as CLINK input.

---

005 `WARNING: Reference not resolved during autocall: [symbol_name]`

Explanation

No definition was found for the symbol [symbol_name], and the symbol could not be resolved by autocall from an AR370 archive. Since the symbol is an extended name, no explicit attempt is made to resolve this symbol from autocall libraries in PDS or TXTLIB format.

---

006 `ERROR: Error while writing to SYSJCLIN file.`

Explanation

An attempt to write one or more linkage editor control statements to the SYSJCLIN output file stream has been unsuccessful. Usually this is caused by having insufficient space available for the file, but any file system problem or failure that might cause **`fwrite()`** to fail could also cause this message.

Action

Re-run CLINK with the =warning run-time option to provide additional SAS/C Library diagnostics. Make sure the space available for the output file is large enough to hold all the output.

---

007 `ERROR: Error occurred while reading an input file.`

Explanation

An attempt to read the next record from an input file that was previously opened has failed. This can be caused by a truncated object file or AR370 archive, but any file system problem or failure that might cause **`fread()`** to fail could also cause this message.

Action

Re-run CLINK with the =warning run-time option to provide additional SAS/C Library diagnostics. Check all input files for validity and integrity. In general input files should be fixed length record format with 80 byte records.

---

008 **`ERROR: Error occurred while reading SYSIN.`**

Explanation

MVS only: An attempt to read the next card from the SYSIN DDname that was previously opened has failed. This may be caused by a truncated object file, but any file system problem or failure that might cause **`fread()`** to fail could also cause this message.

Action

Re-run CLINK with the =warning run-time option to provide additional SAS/C Library diagnostics. Check all input for validity and integrity.

---

009 **`ERROR: Input error occurred. SYNADAF message text follows:`**

Explanation

MVS only: Under certain conditions CLINK uses the SAS/C OS low-level I/O. An input error occurred processing a PDS using BSAM. The SYNADAF macro is used to obtain information about the error. This information is displayed in the LSCL010 message that follows this message.

---

010 **`ERROR: [error]`**

Explanation

MVS only: CLINK displays error information from the SYNADAF macro. If this message includes the text "WRNG.LEN.RECORD," the problem is most likely an input PDS which does not contain card images (such as a load module library) or an incorrect concatenation.

Note that in a concatenation, all concatenated PDS's should have the same RECFM and LRECL, and the file with the largest blocksize should be concatenated first (or alternately an explicit DCB should be specified on the first concatenated DD statement). If the text "WRNG.LEN.RECORD" does not appear, the problem may be a hardware error.

Action

Check the IBM message IOS000I in your job log, and consult your site's systems programming staff to determine whether this is a hardware problem and how it can be corrected.

---

087 `NOTE: Can't open file [filename].`

Explanation

CMS and UNIX only: The first attempt to open the named AR370 archive failed. Processing continues. Usually this is because the specified Archive is not found. Check the spelling of the [filename]. Any file system problem or failure that might cause open() to fail could also cause this message.

Action

Re-run CLINK with the "=warning" run-time option to provide additional SAS/C Library diagnostics.

---

087 `ERROR: Can't open file [filename].`

Explanation

A specified AR370 archive was found and partially processed, but a subsequent attempt to re-open the named AR370 archive failed. Under some circumstances CLINK may temporarily close an archive file, so any other program may then open the file or lock the file; if CLINK then tries to re-open the archive to continue processing this message is produced.

Any file system problem or failure that might cause open() to fail could also cause this message.

Action

Re-run CLINK with the "=warning" run-time option to provide additional SAS/C Library diagnostics.

---

088 `WARNING: Invalid AR370 library specified: [name].`

Explanation

The named file contains an invalid AR370 archive. On some systems a single file or concatenation may contain more than one AR370 archive. In such cases, CLINK processes each archive in order from the beginning of the concatenation to the end or the first invalid archive in the concatenation. Any valid AR370 archives processed prior to the invalid archive are retained for autocall processing.

---

089 `ERROR: [error] occurred performing I/O on AR370 library: [name].`

Explanation

An error described by [error_text] occurred processing the named AR370 archive.

Action

Re-run CLINK with the =warning run-time option to provide additional SAS/C Library diagnostics.

---

100 `ERROR: Can't create object file for CLINK output: [error]`

Explanation

An error described by [error] occurred when creating the output object file. Any file system problem or failure that might cause open() to fail could also cause this message.

Action

Re-run CLINK with the "=warning" run-time option to provide additional SAS/C Library diagnostics.

---

101 `ERROR: Can't open object file: [filename].`

Explanation

One of the primary input files could not be opened successfully. Check the filename and filetype specification on the command line for invoking CLINK.

---

102 `WARNING: Can't open file during autocall: [filename].`

Explanation

Mainframe only: A specific attempt to resolve an external symbol during autocall failed. CLINK attempted to open a file or library member with a name derived from the symbol name. This is only a warning because the reference may be resolved as CLINK autocall processing continues or later by the linker.

---

103 `ERROR: Not enough memory available, CLINK cannot continue.`

Explanation

An attempt to allocate memory failed.

Action

Increase the amount of memory available to CLINK.

---

105 `ERROR: Can't open terminal for input.`

Explanation

CMS only: the primary input file to CLINK is specified on the command line. However, the attempt to open the terminal file failed. See the run-time library messages for more information.

---

106 `WARNING: file name [filename] truncated to [first eight characters of file name].`

Explanation

CMS filenames can be no longer than eight characters.

---

107 `ERROR: No primary input file specified.`

Explanation

CMS only: Primary input files are files named on the command line for the CMS CLINK EXEC. End-of-file on the terminal file was reached before any input was read.

108 `ERROR: SYSJCLIN not allocated.`

Explanation

The open for the SYSJCLIN output file failed. On MVS make sure the DDname is correctly defined.

111 `NOTE: –o output-file option required but not specified.`

Explanation

UNIX only: The output file name must be specified.

Action

Refer to `–o` options in the *SAS/C Compiler and Library User's Guide.*

112 `WARNING: File name [filename] truncated to [first eight characters of file name].`

Explanation

CMS only: See message 106.

113 `ERROR: [parameter] is not a valid CLINK option.`

Explanation

UNIX only: CLINK terminated because the argument [parameter] was not recognized as a CLINK option.

Action

Refer to the CLINK options for the options appropriate for your specific operating system in the *SAS/C Compiler and Library User's Guide.*

114 `ERROR: Can't create print file for CLINK output.`

Explanation

CMS only: `stdout` could not be opened successfully.

Action

Check `stderr` for possible library diagnostics.

117 `WARNING: Invalid DDname prefix specified [prefix].`

Explanation

MVS only: The **`files[xxx]`** option has been specified with an unacceptable prefix [prefix] for DDnames.

124 `WARNING: Prefix truncated to [symbol].`

Explanation

A prefix in a **`GATHER`** control statement (mainframe) or **`-g`** CLINK option (cross-platform) exceeds six characters in length. CLINK will truncate the prefix from the right to six characters.

Action

Change the control statement to remove or replace the prefix.

125 `WARNING: Duplicate prefix [symbol] ignored.`

Explanation

A prefix in a **`GATHER`** control statement is the same as a prefix in the same control statement or a previous one. CLINK ignores every instance of this prefix except the first.

Action

Remove the duplicate prefix from the **`GATHER`** control statement.

200 `ERROR: Load failed for extended names user exit.`

Explanation

The user has specified the ENEXIT option, and an attempt to load the user's exit to supply the external symbol names has failed.

201 `ERROR: Extended names user exit returned code [return_code]`

Explanation

The user exit associated with the ENEXIT option returned a return code other than 0 or 4, so the meaning of [return_code] is undefined. This is an error in the user exit, which causes CLINK to terminate.

202 `ERROR: Extended names user exit ID value out of range: [id_value]`

Explanation

The ID value must not exceed 999999. The user exit associated with the ENEXIT option has supplied an out of range value [id_value] for the integer part of the external symbol name. The value must be between 0 and 999999 inclusive. This is an error in the user exit, which causes CLINK to terminate.

203 `ERROR: Problems executing user exit: [message]`

Explanation

UNIX only: When [message] is "Problems with execl()" the user exit associated with the ENEXIT option is not executable.

Otherwise, [message] corresponds to **`strerror`** errno for the failure of the initialization of the pipes to or from the user exit program or script.

300 `ERROR: Code generator should emit CM or ER immediately following the @EXTERN# SD card. THIS CSECT HAS BEEN IGNORED.`

Explanation

Use of the CSECT name @EXTERN# by non SAS/C code interferes with the library's functioning. Code compiled with the SAS/C Compiler should not generate this message.

301 `ERROR: multiple initialization sections for [csect name].`

Explanation

Two modules containing initializations of external variables have the same section name. Details on how section names are assigned are provided with the discussion of the **`sname`** option in the *SAS/C Compiler and Library User's Guide, Fourth Edition.* This error message may result from using CLINK to

- process more than one module compiled with the same **`sname`**
- process a module compiled without an **`sname`** in which the first function of the module is the same as the **`sname`** given to another module
- process two modules compiled without specifying an **`sname`** in which the name of the first function in the modules is the same
- link more than one module that contains only static data/functions.

Note: For a compilation containing no external functions, the name of the first defined external is used as the default sname.

602 `WARNING: symbol initialized twice: [external name].`

Explanation

Two modules being linked contain initializations for the same external variable.

603 `WARNING: multiple occurrences of [routine name] routine in library.`

Explanation

Two CSECTs with the same name were found in different members of a library or archive used for autocall. Which CSECT will be present in the output is unpredictable.

Action

Make sure that an autocall library or archive does not contain several copies of the same CSECT.

604 **`WARNING: GATHER output table [symbol] multiply defined.`**

Explanation

A label definition (LD) or named control section (SD) with the same name as the **`GATHER`** table was found in an object module. The linkage editor or loader ignores the **`GATHER`** table with this name. Therefore, the program will probably execute incorrectly.

Action

Remove or rename the LD or SD, or change the prefix to one that does not cause a conflict.

---

700 **`NOTE: Control statement not processed by CLINK, passed on to linker.`**

Explanation

The control statement was recognized by CLINK but was not processed. The control statement is copied to CLINK output for use by the mainframe linkage editor or loader.

---

701 **`WARNING: Unrecognized control statement passed on to linker.`**

Explanation

CLINK encountered a control statement that it did not recognize. The control statement is copied to CLINK output for use by the mainframe linkage editor or loader.

---

702 **`ERROR: Can't open INCLUDE file: [filename].`**

Explanation

A file specified in an **`INCLUDE`** statement could not be opened successfully.

Action

Check the file specification for the file.

---

703 **`ERROR: invalid syntax in INCLUDE statement.`**

Action

Check the *SAS/C Compiler Library and User's Guide* for the proper syntax for control statements.

---

705 **`ERROR: invalid syntax in [operation] statement.`**

Explanation

A control statement that specifies an [operation] of either INCLUDE, GATHER, or ARLIBRARY, has incorrect syntax. These CLINK control statements should have the same general format, and obey most of the same conventions as linkage editor control statements.

---

706 `ERROR: Multiple NAME control cards in linkage. Execution terminated.`

Explanation

The input to CLINK contained more than one NAME statement. The NAME control statement identifies the end of the input for an output load module to the linkage editor. Since CLINK only supports a single output load module, this usage is not valid for CLINK.

707 `WARNING: continuation of control statements not supported.`

Explanation

Column 72 of any control statement must be blank because CLINK does not support continuation of control statements.

802 `WARNING: pseudoregister data truncated`

Explanation

The pseudoregister offset does not fit into the space provided for that value.

Note: This message should only occur for code not produced by the SAS/C Compiler. If the code is assembler code, rewrite it to use longer Q-type address constants. If it is PL/I code (or code from another language that uses pseudoregisters), do not input the object code to CLINK; instead, input it to the linker. See the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for information on linking programs using SAS/C with another high-level language.

804 `ERROR: No TXT card for storing CXD information from RLD.`

Explanation

A CXD in the relocation dictionary was not processed. This problem can only occur in malformed object decks.

805 `ERROR: No TXT card for storing XD information from RLD.`

Explanation

An XD in the relocation dictionary was not processed. This problem can only occur in malformed object decks.

806 `NOTE: NO PSEUDOREGISTERS IN THIS LINKAGE`

Explanation

The **`PREM`** CLINK option has been specified, but there are no pseudoregisters to be processed. (This message is for information only.)

811 `ERROR: Expected CM or ER after @EXTERN# SD not found. THIS CSECT WILL BE IGNORED.`

Explanation

The first card after the @EXTERN# ESD card must be an ESD of type ER or type CM which refers to the code CSECT which generated the @EXTERN#. This problem can only occur in malformed or corrupted object decks.

812 `WARNING: Unmatched SMP library reference [lib_id] in [csect_name].`

Explanation

The SMP distribution library identifier lib_id is not matched with an element identifier. This only occurs if the SAS/C SMP Support Libraries have been corrupted.

Action

Call SAS/C Technical Support.

813 `WARNING: Unmatched SMP element reference [element] in [csect_name].`

Explanation

The SMP distribution library element identifier element is not matched with a SMP distribution library identifier. This only occurs if the SAS/C SMP Support Libraries have been corrupted.

Action

Call SAS/C Technical Support.

814 `WARNING: Multiple SMP library references in [csect]. ''[libname]'' replaces ''@##0[lib_id]''.`

Explanation

This only occurs if the SAS/C SMP Support Libraries have been corrupted.

Action

Call SAS/C Technical Support.

815 **`WARNING: Multiple SMP element references in [csect]. ``[element]'' replaces ``@##[element_id]''.`**

Explanation

In the CSECT named [csect], an SMP distribution library element identifier "@##[element_id]" has not been matched with a library identifier and is being replaced by another element id [element]. This element identifier may not appear in the SYSJCLIN output file. This only occurs if the SAS/C SMP Support Libraries have been corrupted.

Action

Call SAS/C Technical Support.

---

818 **`ERROR: The SNAME ``[section_name]'' is multiply defined.`**

Explanation

Two or more input object decks have the same section name. During compilation the compiler creates names for various CSECTs based on the section name. In order to resolve references to these CSECTs, they must have unique names. This can be accomplished by specifying different SNAMEs for each compilation.

Action

Refer to the *SAS/C Compiler and Library User's Guide* for more on **`sname`** compiler options and default section names.

---

6 LSCL COOL Messages

This chapter documents messages generated by the COOL utility (either mainframe or cross-platform). Each message contains the message text and an explanation of the message, as well as operating-system information and the action needed to correct the error, if applicable. All messages can occur in any operating environment unless otherwise indicated.

COOL diagnostic messages have the form

```
LSCL[num] [severity]: [message text]
```

On UNIX, they have the form

```
cool: [severity] [num]: [message text]
```

where

num	is the message number
severity	NOTE, WARNING, or ERROR
message text	is a description of an action (NOTE), possible problem (WARNING), or condition that will prevent further processing (ERROR).

Messages are written to **stdout** if the **term(-t)** option is in effect; they are written to the listing if a listing is created. If a WARNING message is generated, the final return code from COOL will be no less than 4. If an ERROR message is generated, COOL will terminate immediately with a return code of 8.

Messages 1972–1979 describe errors generated by the COOL driver program, COOLB. These messages are always written to **stderr** even if **NOTERM** is specified. They are not written to the listing.

Note: The COOL message prefix LSCL has been omitted from the descriptions below.

LSCL Messages

1000 ERROR: There is not enough storage to continue.

Explanation

COOL attempted to allocate more storage but the allocation failed.

Action

If you are running COOL in MVS batch, increase the size of the region using the REGION= parameter on the JOB or EXEC statement in your JCL. If you are running COOL in TSO, increase your TSO region size. If you are running in CMS, increase the size of your virtual machine.

Normally, COOL keeps all its data in storage until it has processed all the input files. If you are processing very large groups of object files in a 370-mode virtual machine, COOL may need more storage than is available. Note, this will increase the length of time require to process the object files. If you are running COOL in MVS batch, you will probably need to increase the amount of time allocated to the job.

1001 `WARNING: The option ``[option]'' is undefined.`

Explanation

The option specified by [option] is not defined by COOL.

Action

The option may be misspelled. Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options. Re-run COOL with the corrected option.

1002 `ERROR: Can't [open | close | write to] the [input | included | output | listing] file ``[name].'' [The reason given was ``reason'']`

Explanation

An error occurred while processing the input, output, or listing file identified by [name]. In MVS, TSO, and CMS, this message will be accompanied by a run-time library diagnostic message containing more detail about the condition. In UNIX, [reason] is the **`strerror`** information associated with the error.

Action

Correct the condition and re-run COOL.

1003 `ERROR: The output file name is required but was not specified.`

Explanation

UNIX only: Use the **`-o`** option to specify the name of the output file.

1004 `ERROR: There were no input files specified.`

Explanation

No input object files were specified on the command line.

Action

Re-run COOL, including at least one object file name on the command line.

1005 **ERROR: Record [num] in the input object file ``[name]'' has an unsupported format. A data display follows:**

Explanation

COOL encountered a record that is not a valid object record. [name] is the file in which the record occurs, and [num] is the record number. This occurs when a file that is not an object file is specified as input or, rarely, when an object file contains non-standard records.

Action

Make sure that all the input files are object files. If the file is an object file but contains non-standard records, refer to the documentation for the program that produced the file for information about how to use the file. If the file is an object file produced by the compiler or C++ translator, report this message to SAS/C Technical Support.

Since object records are in binary format, COOL displays the contents of the record in a formatted text display that looks like this:

```
.TXT ... .. ...00.. MAIN(FFIND)
.........................,.....0...J........
0EEE4000440344004FF214444DCCD4CCCDC5440300010000000000009ED0155EC04FE
237300000008000170081000041 95D66954D0002001800000000000A0C8F8004700
```

1006 **ERROR: No control section is defined for ESD item [item] in record [num] in input object file ``[name].'' A data display follows:**

Explanation

COOL encountered a malformed object record in the file identified as [name] at record number [num]. The object file may have been corrupted by an editor or have been partially overwritten by another program.

Action

Re-compile the source file and re-run COOL. If the problem persists, contact SAS/C Technical Support. Refer to message 1005 for a description of the data display.

1007 **ERROR: Text record [num] in the input object file ``[name]'' refers to a non-existent ESDID item. A data display follows:**

Explanation

Refer to the explanation in message 1006.

1008 **ERROR: The RLD item at data offset [num] in record [num] in the input object file ``[name]'' refers to a non-existent [position | relocation] ESDID item.**

Explanation

Refer to the explanation in message 1006.

1009 `ERROR: The section name ''[sname]'' occurs in both the ''[name1]'' and ''[name2]'' input object files.`

Explanation

Both of the named object files were assigned the section name [sname] by the compiler. Since the compiler uses the section name to create some external symbols, this can cause problems when linking. This problem can occur when the name of the first external function becomes the default section name, and both have the same initial seven characters.

Action

Re-compile the source files using the compiler's **`sname`** option to assign a unique section name to each file and re-run COOL. In the extremely rare case where more than one object file must have the same section name, use the DUPSNAME option to suppress this message.

1010 `NOTE: Pre-linking completed with return code = [num]`

Explanation

COOL completed processing the input files. The final return code was [num].

1011 `ERROR: An invalid [INCLUDE | INSERT | ARLIBRARY | control | continuation of control] statement was detected. A data display follows:`

Explanation

A control statement of the specified type was found in an input file.

Action

Refer to the explanation of message 1005 for a description of the data display. Correct the statement and re-run COOL.

1012 `ERROR: Unexpected end-of-file, or the file ''[name]'' is not a valid ar370 archive.`

Explanation

This message is issued when COOL tries to read information from the named ar370 archive, but the archive is too small. This is usually caused by a corrupted archive.

Action

Re-build the archive and re-run COOL.

1013 **`ERROR: There is a duplicated definition of ``[symbol]'' in input object files ``[name1]'' and ``[name2].''`**

Explanation

Both of the named input object files contain a definition of the external symbol [symbol]. This is usually caused by defining an external symbol in more than one compilation in a program that is not re-entrant. Only one compilation in a program should define an external symbol.

Action

Correct the problem and re-run COOL.

1014 **`WARNING: The input object file ``[name]'' contains duplicate initialization data for symbol ``[symbol]''`**

Explanation

The named input object file contains initialization data for the external symbol [symbol], but COOL has already processed initialization data for this symbol in a previous input file. This is usually caused by defining an external symbol in more than one compilation in a re-entrant program. COOL discards the second (and subsequent) initial values, but assigns the strongest alignment and longest length of all the initial values. Only one compilation in a program should define an external symbol.

Action

Correct the problem and re-run COOL.

1015 **`NOTE: The reference to ``[symbol]'' was resolved from ``[name]''`**

Explanation

During autocall processing, COOL found the definition of the named symbol when it included file [name]. This message is for information only, and is emitted only when the VERBOSE option is used.

1016 **`WARNING: The file ``[name]'' was included but did not define ``[symbol]''`**

Explanation

During autocall processing, COOL tried to resolve external symbols by reading a file whose name was the same as the symbol name. COOL attempted to resolve the external reference symbol by including file [name]. However, file [name] did not contain a definition for symbol. If [name] is supposed to contain the definition for [symbol], check the source file to determine why [symbol] is not defined.

1017 `WARNING: The reference to ''[symbol]'' in ''[name]'' is unresolved.`

Explanation

COOL completed autocall processing but did not find a definition for the external reference symbol found in file [name]. This does not necessarily indicate a problem if the definition will be supplied during subsequent linking, or if the reference is never used when the program executes. Otherwise, the reference may be misspelled, or the object file that contains the definition was omitted from the COOL input files.

When running under UNIX, this message is a NOTE if the **LET** or **-m** option is specified.

1018 `WARNING: In the ''[csect]'' CSECT in input object file ''[name]'' the ''[symbol]'' pseudoregister displacement at offset [num] has been truncated.`

Explanation

The **PREM** option is in effect. As COOL removed pseudoregisters from the output object file, it determined that object file [name] defined a Q-type address constant for [symbol] having a length too small to contain the pseudoregister offset. For example, the offset is greater than 255, but the address constant is defined as QL1. This can only occur in object files created from assembler language source files. [num] is the offset from the start of the CSECT at which the address constant occurs.

Action

Increase the size of the address constant, re-assemble the source file, and re-run COOL.

1019 `ERROR: ''[name]'' is not the name of a CMS file.`

Explanation

CMS only: The string [name] was used as the name of an input file, output file, or listing file, but the string cannot be used to form the name of a CMS minidisk or Shared File System file.

Action

Consult *SAS/C Library Reference, Volume I*, for an explanation of how these fileids are formed.

1020 `NOTE: No pseudoregisters were found in the input files.`

Explanation

The **PREM** option was specified but no pseudoregisters were found in the input files. This message is for information only.

1021 **WARNING: The ar370 archive ''[name]'' contains data inconsistent with the actual file. This file will be ignored.**

Explanation

This message is issued when COOL detects a pointer in the header of the named ar370 archive that is outside the bounds of the archive itself. This is caused by a corrupted archive.

Action

Re-build the archive and re-run COOL.

1022 **WARNING: Only ar370 archives preceding this one will be used.**

Explanation

MVS/TSO only: This message is only issued in conjunction with messages 1012 and 1021 when the **CONTINUE** option is used. The corrupted ar370 archive is a member of a concatenation, following at least one uncorrupted archive. COOL will continue processing using the uncorrupted archives in the concatenation, but will ignore the data in the corrupted archive and any archives that may follow it in the concatenation.

1023 **ERROR: Some text data is missing in the extended names CSECT ''[csect]'' in file ''[name].''**

Explanation

COOL encountered a malformed object record in the extended names CSECT [csect] in the file identified as [name]. The object file may have been corrupted by an editor or have been partially overwritten by another program.

Action

Re-compile the source file and re-run COOL. If the problem persists, contact SAS/C Technical Support.

1024 **NOTE: The ''[type]'' control statement was not processed, and was copied to the output file.**

Explanation

COOL encountered a linkage editor control statement (such as **IDENTIFY** or **CHANGE**) in one of the input files. COOL simply copies the statement to the output file. This message is for information only.

1025 **NOTE: [Beginning | End of] ''[name]'' processing.**

Explanation

This message is only issued when the **VERBOSE** option is specified and COOL has started or finished processing an input file.

1026 **`WARNING: The specified DDname prefix contains invalid characters: ``[prefix]''`**

Explanation

MVS/TSO only: The DDname prefix [prefix] specified by the **`FILES`** option is not a valid DDname prefix. The prefix must be 1 to 3 characters, and the first character must be alphanumeric or $, @, or #. The remaining characters must be alphanumeric.

1027 **`WARNING: The value specified for the ``[-p | -zf]'' option is not in the range of acceptable values.`**

Explanation

Either the **`pagesize`** (**`-p`**) or **`maxfiles`** (**`-zf`**) option specified a value that was out of range for that option. The value of the **`pagesize`** option must be an integer greater than or equal to 5 or -1 to suppress pagination. The **`maxfiles`** option must be a positive integer.

1028 **`WARNING: The file ``[name]'' could not be opened for autocall.`**

Explanation

During autocall processing, COOL tried to open the file [name] to resolve an external symbol but could not. If another input file defined the symbol that COOL was trying to resolve, this is not a problem. However, if this file is required to resolve the symbol, ensure that the file contains the symbol definition and is available to COOL as either primary or secondary input.

1029 **`ERROR: A BSAM error occurred while processing the file ``[name].'' The following message(s) were passed to the error analysis routine: [message1] [message2]`**

Explanation

MVS only: COOL handled a BSAM error while opening or reading the named input file. BSAM called the error analysis routine with the information displayed as [message1]. If the input file is a member in a PDSE, additional information about the error is displayed as [message2]. This message may indicate a hardware failure, or it may indicate that the named file is not concatenated correctly.

Action

If the named file is a concatenation, verify that the concatenated files all have the same DCB characteristics. If the problem is not an invalid concatenation, report the problem to your site's System Programming staff for assistance.

1030 **`ERROR: An unexpected end-of-file was encountered after record [num] in the input object file ''[name]''`**

Explanation

COOL processed a control statement that had a non-blank character in column 72, indicating that the control statement was continued in the next record. However, the input file ended after the current record.

Action

There may be a spurious non-blank character in column 72 of the control statement. Remove it and re-run COOL.

1031 **`ERROR: The continuation of the ''[keyword]'' control statement at record [num] in the input object file ''[name]'' is not a valid control statement. A data display follows:`**

Explanation

COOL processed a control statement that had a non-blank character in column 72, indicating that the control statement was continued in the next record. However, the next record in the input file was not a control statement.

Action

There may be a spurious non-blank character in column 72 of the control statement. Remove it and re-run COOL. Refer the explanation of message 1005 for a description of the data display.

1032 **`ERROR: The ''[keyword]'' control statement at record [num] in the input object file ''[name]'' is not in the correct format.`**

Explanation

The specified control statement has a syntax error.

Action

Correct the error and re-run COOL.

1033 **`ERROR: In the continuation of the ''[keyword]'' control statement at record [num] in the input object file ''[name]'', columns 1 through 15 must be blank.`**

Explanation

The second and subsequent record parts of a continued control statement must begin in column 16 or after of the continuation record.

Action

Correct the error and re-run COOL.

1034 **WARNING: An INCLUDE statement was found at record [num] in the included object file ``[name].'' Data following this record are not processed.**

Explanation

COOL has encountered an **INCLUDE** statement in the included file name. This message is a reminder that no data following the **INCLUDE** statement will be processed.

By default, COOL handles **INCLUDE** statements in included files the same way the IBM linkage editor handles them. That is, when an included file contains an **INCLUDE statement**, the specified module(s) are also included, but any data following the **INCLUDE** statement is ignored.

Action

This behavior can be overridden by specifying the **noinceof** option (in MVS/TSO or CMS) or the **-zi** option in UNIX.

1035 **ERROR: The input object file ``[name]'' can't be reopened. The maxfiles limit has been reached and no files can be quiesced.**

Explanation

The specified file was temporarily closed (quiesced) but cannot be reopened. COOL uses the value of the **maxfiles** option to determine how many input files to have open simultaneously. This number has been reached, so COOL temporarily closed one of its input files. However, this file cannot be reopened.

Action

Re-run COOL, specifying a larger value for **maxfiles**. If the problem persists, contact SAS/C Technical Support.

1036 **ERROR: An unknown error occurred while reopening ``[name].''**

Explanation

The specified file was temporarily closed (quiesced) but an error occurred when COOL tried to reopen it. The file may have been erased by another program.

Action

Look for operating system messages or run-time library diagnostics that may help explain the problem. If the problem persists, contact SAS/C Technical Support.

1037 **WARNING: The ``[prefix1]'' GATHER prefix is longer than 6 characters. It will be truncated to ``[prefix2]''**

Explanation

GATHER prefixes may be no longer than 6 characters. COOL will truncate [prefix1] to the value shown as [prefix2].

1038 **`WARNING: A symbol matching the GATHER table name ''[symbol]'' is defined in input file ''[name]''`**

Explanation

An external symbol having the same name as a **`GATHER`** table is defined in the input file name. This is likely to cause the program to fail when it is executed.

Action

Change the symbol name so that the **`GATHER`** table name is unique to the input files.

1039 **`WARNING: The duplicated ''[prefix]'' GATHER prefix will be ignored.`**

Explanation

The prefix was specified more than once in the input files. The second and subsequent specifications are ignored.

1040 **`WARNING: The inserted symbol ''[symbol1]'' is longer than 8 characters and will be truncated to ''[symbol2]''`**

Explanation

Symbols in **`INSERT`** control statements must be no more than eight characters long. The symbol [symbol1] will be truncated on the right to the symbol shown as [symbol2].

Action

If this is not correct, change the **`INSERT`** control statement and re-run COOL.

1042 **`WARNING: SMP identifiers distribution library and element id are not matched.`**

Explanation

SAS/C SMP Supporting Libraries are incorrect.

Action

Contact SAS/C Technical Support.

1044 `ERROR: Problem with extended names user exit: [text]`

Explanation

The value for [text] provides the explanation, from the following list:

Problem setting pipes for exit process.

Problem running exit process.

Problem with dynamic load of User Exit.

Non-zero return code from User Exit.

User Exit returned id out of range [message].

Action

Contact SAS/C Technical Support.

1045 `Error: Trouble reading shelled object [part] : [file].`

Explanation

[file] specifies the file in error; [part] represents one of the following:

header	shelled object header not read.
header size	shelled object header size not read.
symbols	shelled object symbol information not read.
references	shelled object template reference introduction not read.
strings	shelled object string table not read.

Difficulties (such as an invalid object deck) were encountered while reading a shelled object.

Action

Contact SAS/C Technical Support.

1046 **`Trouble [action] output debugger file ([file]), ignored.`**

Explanation

[file] specifies the file in error; [action] represents one of the following:

opening	The output debugger file could not be opened for output.
writing header to	The output debugger file could not be written to.
writing	The output debugger file could not be written to because of a possible dataset size problem.

An error was encountered while writing a debug file. COOL performs this activity only for objects which include debugging information. File creation problems typically produce this error.

Action

Increase the space that is available in the output destination, or change that destination with the COOL DBGLIB option.

1047 **`Trouble [action] temporary debugging file, ignored.`**

Explanation

[action] indicates one of the following:

opening	Problem was encountered opening the temporary file.
writing	Problem was encountered writing the temporary file.

A temporary file is allocated to hold intermediate versions of all debugger files that were included in the objects. This file can be very large.

Action

On MVS, a dataset definition can be made to increase the storage available for this file. On CMS, this file is allocated on the minidisk with the most writable space.

1048 *sev*`: Input file ''`*`filename`*`'' previously prelinked.`

Explanation

During the processing of input object file filename, COOL detected that it had previously processed this object file, and had marked it as not to be reprocessed.

If the **`IGNORERECOOL`** option was specified, the mark to reprocess the object file does not cause an error and processing continues. If both the **`IGNORERECOOL`** and **`VERBOSE`** options were specified, the message is displayed with the severity *sev* level of **`NOTE`** and processing continues.

If the **`NOIGNORERECOOL`** option was specified or if it was the default, the message is displayed with a severity *sev* level of **`ERROR`**, and processing stops with a return code of 8.

This message is also issued if COOL cannot locate the extended name in an input object deck. The reason that COOL cannot locate the extended name is that the input object deck has already been prelinked.

COOLB Diagnostic Messages (MVS/TSO Only)

The following messages are produced by the COOL driver program, COOLB. They are always written to **stderr** even if **NOTERM** is specified; they are not written to the listing.

1972 **WARNING: Option ''[option]'' begins with a parenthesis.**

Explanation

No COOL option can begin with a parenthesis. This error is probably caused by inserting a blank between an option and its parenthesized argument list.

Action

Remove the intervening blank(s) and re-run COOL.

1973 **WARNING: For option [option], [text1] was expected but [text2] was found.**

Explanation

COOLB detected a syntax error in the specification of [option].

Action

Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for the syntax of COOL options. Correct the error and re-run COOL.

1974 **WARNING: Option [option] does not take arguments.**

Explanation

[option] was followed by a parenthesized argument list, but this option does not accept any arguments.

Action

Remove the argument list and re-run COOL. Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options.

1975 **WARNING: Option [option] expects arguments.**

Explanation

[option] requires a parenthesized argument list. For example, the **pagesize** option must be specified as **pagesize(nn)**, where **nn** is the number of lines to be printed on a listing page.

Action

Add the necessary argument and re-run COOL. Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options.

1976 `WARNING: Option [option] cannot be negated.`

Explanation

Option is prefixed with "`no`," but option cannot be negated. For example, it is not possible to specify `nopagesize` as an option.

Action

Remove the `NO` prefix and re-run COOL. Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options.

1977 `WARNING: Option [option] is not recognized.`

Explanation

[ption] is not a COOL option.

Action

Remove the word from the parameter list and re-run COOL. Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options.

1978 `WARNING: Option abbreviation is ambiguous ([option1] vs [option2]).`

Explanation

COOL options may be abbreviated to the fewest unambiguous characters, but abbreviation is too short to clearly indicate which COOL option is intended. The abbreviation could mean either [option1] or [option2].

Action

Extend the abbreviation enough to indicate which option is intended and re-run COOL.

1979 `WARNING: Linking was suppressed. [1 error was | [num] errors were] encountered.`

Explanation

This message is issued when at least one error is diagnosed in the options list. COOLB does not invoke COOL or the system linker.

Action

Refer to the *SAS/C Compiler and Library User's Guide, Fourth Edition* for a list of COOL options. Correct the error and re-run COOL.

7 LSCP CICS Preprocessor Messages

This chapter describes diagnostic messages generated by the CICS Preprocessor. Once any preprocessor errors have been corrected, your program should be ready to compile and link-edit.

CICS diagnostic messages have the form

```
LSCL[num] [severity]: [message text]
```

where [severity] can be any of the following:

NOTE	provides information about expected behavior. The return code is not changed.
WARNING	provides information about conditions that will not prevent translation from completing successfully. The return code is set to 4.
ERROR	provides information about conditions that will prevent translation from continuing. If an error occurs before translation begins (such as a file I/O error), the preprocessor will terminate immediately. If an error occurs after translation has begun, the preprocessor will stop translating, but will check all the remaining commands in the input file for correct syntax. The return code is set to 8.
SEVERE [Error]	provides information about conditions that will cause immediate termination. The return code is set to 12. Diagnostic messages usually change the return code set by the preprocessor.
no level	message text the same for both an error and a warning.

Note: The LSCP prefix has been omitted in the message descriptions below.

LSCP Messages

000 **`SEVERE: Not enough storage to continue.`**

Explanation

An attempt to allocate storage failed.

Action

Increase the virtual machine or region size.

001 `WARNING: Unknown option [option] ignored.`

Explanation

The specified option was not a preprocessor option.

Action

The option may have been misspelled.

---

002 `WARNING: Keyword parameter expected for option [option]. Option ignored.`

Explanation

A required parameter was not given for the specified option.

---

003 `WARNING: [option] is not a keyword option. Option ignored.`

Explanation

A parameter was used on an option that does not require one.

---

004 `WARNING: [option] may not be negated. Option ignored.`

Explanation

An option, such as **`pagesize`**, was incorrectly preceded by **`no`**, but the option may not be negated.

---

005 `SEVERE: Unable to open the [file] file.`

Explanation

The specified required file could not be opened.

Action

MVS and CMS only: refer to the run-time library messages on **`stderr`** for more information about why the file could not be opened.

---

006 `SEVERE: Error occurred while reading from the [file] file.`

Explanation

The specified required file could not be used for input any longer.

Action

MVS and CMS only: refer to run-time library messages on **`stderr`** for more information about the error.

---

007 **`SEVERE: Error occurred while writing to the [file] file.`**

Explanation

The specified required file could not be used for output any longer.

Action

MVS and CMS only: refer to run-time messages on **`stderr`** for more information about the error.

---

008 **`WARNING: Invalid DDname prefix [prefix] specified by FILES option. DDname prefix set to default value.`**

Explanation

MVS only: The specified prefix is longer than three characters or contains a character that may not appear in a DDname.

Action

Correct the prefix. The prefix may be no more than three characters in length. The first character must be alphabetic or @, #, or $. The second and third characters must be alphanumeric or @, #, or $.

---

009 **`WARNING: Value of pagesize option, [value] is not numeric or is less than 7. Page size is unchanged.`**

Explanation

The **`pagesize`** option requires an integer parameter greater or equal to 7.

Action

`pagesize`(7) will cause the source listing to be printed with only one line per page.

---

010 **`WARNING: No value given for option [option]. Option ignored.`**

Explanation

A required parameter was not given for the specified option.

---

011 **`SEVERE: Cannot access character translation table.`**

Explanation

The custom character translation table could not be loaded.

Action

This is possibly due to an out-of-storage condition. MVS and CMS only: Refer to run-time messages on **`stderr`** for more information about the error. Tell your SAS Software Representative for C Compiler Products about the error.

---

012 `WARNING: Invalid [option] specification [value]. Option ignored.`

Explanation

The parameter for the specified option was incorrect or out of range.

013 `SEVERE: Unexpected end of file on input file.`

Explanation

An end-of-file condition occurred while a comment, string literal, character literal, CICS, or DL/I command was being processed.

014 `WARNING: Input line length exceeds output line length.`

Explanation

A line in the input file was longer than may be contained in a single line in the output file.

Action

On MVS, increase the output file LRECL. On CMS or UNIX, split the line in the input file into several shorter lines.

015 `SEVERE: Error occurred while closing the [file] file.`

Explanation

The specified file could not be closed. If the file is the output file, it may not be usable.

Action

MVS and CMS only: refer to run-time messages for more information about the error.

016 `SEVERE: Cannot access translation data for [command-set] commands.`

Explanation

`command-set` is either CICS or DL/I. An error occurred when the preprocessor attempted to dynamically load the translation data load module.

Action

MVS and CMS only: This is usually caused by not having enough storage available to load the data. If the run-time messages on **`stderr`** indicate that the problem was due to a lack of available storage, increase the virtual machine or region size and re-execute the preprocessor. Otherwise, refer the problem to your SAS Software Representative for C Compiler Products.

017 `WARNING: Right parenthesis assumed.`

Explanation

A command option argument contains a semicolon. The preprocessor assumes that the command has ended and inserts a right parenthesis immediately before the semicolon. The command may not be translated correctly. The compiler may issue error messages for the output source file.

Action

Ensure that all option arguments are properly enclosed in parentheses.

---

018 `WARNING: Semicolon assumed.`

Explanation

A command contains a token that is not an option or an argument.

Action

This warning is usually caused either by a missing semicolon or by a misspelled command option, such as an option that contains lowercase characters. Correct the command by inserting a semicolon at the end of the command or by correcting the misspelled option.

---

019 `ERROR: No command function specified.`

Explanation

A CICS command was terminated by a semicolon before the command function was specified.

Action

This error usually occurs in conjunction with message 018 if the function contains characters that are not allowed in CICS commands. Correct the command before reinvoking the preprocessor.

---

020 `ERROR: Unknown [command-set] function.`

Explanation

`command-set` is either CICS or DL/I. The named function was not a valid function.

Action

The function name may have been misspelled. Correct the command before reinvoking the preprocessor.

---

021 `ERROR: Argument is not associated with an option.`

Explanation

Two successive parenthesized arguments were found. Each argument must be preceded by an option.

Action

Correct the command before reinvoking the preprocessor.

---

022 `ERROR: Expected [option] argument not specified.`

Explanation

The named option requires an argument, but the argument was not found.

Action

Correct the command before reinvoking the preprocessor.

---

023 `ERROR: Unexpected argument after [word].`

Explanation

A parenthesized argument was found where it was not expected.

Action

Correct the command before reinvoking the preprocessor.

---

024 `ERROR: [option] required but not specified.`

Explanation

A required option was not found in this command.

Action

Correct the command before reinvoking the preprocessor.

---

025 `WARNING: [option] argument ignored. [option] assumed.`

Explanation

A parenthesized argument was found following an option that does not accept an argument. The option is accepted as if no argument were used.

Action

Correct the command before reinvoking the preprocessor.

---

026 `ERROR: [option] argument missing.`

Explanation

The named option requires an argument, but one was not specified.

Action

Correct the command before reinvoking the preprocessor.

---

027 `ERROR: [option-a] is valid only if [option-b] is also specified.`

Explanation

The first named option may be used only in conjunction with the second named option.

Action

Correct the command before reinvoking the preprocessor.

---

028 `WARNING: [option] is superfluous or unknown. Ignored.`

Explanation

The named option was either used elsewhere in this command or was not a valid option for this command.

Action

Correct the command before reinvoking the preprocessor. Note that if the option has been specified more than once, the identified instance may not necessarily be the first occurrence.

---

029 `NOTE: No commands translated.`

Explanation

No CICS or DL/I commands were translated. This can occur if the **`nocics`** or **`nodli`** options were specified but CICS or DL/I commands were used. If this occurs, message 034 will be issued for each command in the file.

Action

Determine if the **`nocics`** or **`nodli`** options have been incorrectly specified either when the preprocessor is invoked or in a **`#pragma options`** statement.

---

030 `WARNING: More than 16 options specified. [option] ignored.`

Explanation

No more than 16 options can be specified in a single occurrence of this command. Subsequent options are ignored. This message is issued for each option after 16.

Action

Divide the command into two or more commands.

---

031 `NOTE: ''nohandle'' assumed for GDS command.`

Explanation

All EXEC CICS GDS commands execute as if the **`nohandle`** option is in effect.

---

032 `ERROR: [option-a] is required when [option-b] is used.`

Explanation

The use of one option is dependent upon use of another.

Action

Correct the command before reinvoking the preprocessor.

---

033 `ERROR: One of [option-list] must be specified.`

Explanation

One or more of the options named is required in this command.

Action

Correct the command before reinvoking the preprocessor. Note that this message may not list all of the possible choices.

---

034 `WARNING: EXEC [command-set] found but [NO-command-set] was specified.`

Explanation

A CICS or DL/I command was found in the input file, but the **`nocics`** or **`nodli`** option is currently in effect.

Action

Determine if the **`nocics`** or **`nodli`** option was incorrectly specified when the preprocessor was invoked or in a **`#pragma options`** statement.

---

035 `NOTE: Default [option] supplied.`

Explanation

The preprocessor created a default value for the named option using the **`sizeof`** operator. This message occurs when the DESTIDLENG, FROMLENGTH, KEYLENGTH, LENGTH, PFXLENG, TOLENGTH, or VOLUMELENG options are not coded explicitly in the command. The default value created by the preprocessor may be incorrect, depending on the type of the value used as the operand to **`sizeof`**.

Action

If the default value is correct, no change is needed. Otherwise, specify the correct length explicitly in the command.

---

036 `ERROR: String or character literal at line [line,] column [col] not terminated.`

Explanation

End of file occurred before the string or character literal beginning in column [col] of line [line] in the input file was terminated.

Action

Correct the input file before reinvoking the preprocessor.

---

037 `ERROR: Comment at line [line,] column [col] not closed.`

Explanation

End of file occurred before the comment beginning in column [col] of line [line] in the input file was terminated.

Action

Correct the input file before reinvoking the preprocessor.

---

038 `WARNING: Unknown option ignored.`

Explanation

The indicated option was not valid for this command.

Action

Correct the command before reinvoking the preprocessor.

---

039 `ERROR: [Execute] keyword found. Probable missing semicolon.`

Explanation

Either the EXEC or EXECUTE keyword was found when a command option was expected.

Action

This is usually caused by a missing semicolon between two commands. Correct the input file before reinvoking the preprocessor.

---

040 `ERROR: [option-a] may not be used when [option-b] is used.`

Explanation

[option-a] and [option-b] are mutually exclusive options for this command.

Action

Correct the input file before reinvoking the preprocessor.

---

041 `ERROR: Unable to supply default [option].`

Explanation

This message is issued when the FROM option is omitted in a SEND MAP command, or when the INTO option is omitted in a RECEIVE MAP command. The preprocessor will create a default value for these options if the argument to the MAP option is a string literal. If the argument is not a string literal, this message is issued.

Action

Change the MAP argument to a string literal, or add the INTO or FROM option to the command.

---

042 `ERROR: Missing or invalid [option] argument.`

Explanation

The argument to DFHRESP or DFHVALUE is not one of the defined arguments.

Action

The argument may have been misspelled. Refer to the CICS documentation at your site for a list of defined DFHRESP and DFHVALUE arguments.

---

043 `ERROR: [option-a] may not be used when [option-b] is used.`

Explanation

The named options were incompatible in this command.

Action

Correct the command before reinvoking the preprocessor.

---

044 `ERROR: This command may have undiagnosed syntax errors.`

Explanation

The preprocessor has diagnosed a previous error for this command that prevents the command from being completely analyzed.

Action

Correct the command syntax and reinvoke the preprocessor. Inspect the diagnostic messages to determine if there were any other syntax errors.

---

045 `WARNING: Expecting parenthesized option(s).`

Explanation

A **`#pragma options xopts`** statement was found but no parenthesized list of options follows.

Action

Correct the **`#pragma options xopts`** options before reinvoking the preprocessor.

---

046 `WARNING: Unknown option [option] ignored.`

Explanation

A **`#pragma options xopts`** statement contains the named option, which is not an the preprocessor option.

Action

The option may have been misspelled.

---

047 `WARNING: Keyword parameter expected for option [option]. Option ignored.`

Explanation

The named option, appearing in a **`#pragma options xopts`** statement, requires a parameter, but one was not given.

Action

Add the parameter to the option.

---

048 `WARNING: [option] may not be negated. Option ignored.`

Explanation

The named option, appearing in a **`#pragma options xopts`** statement, may not be negated by adding the **`no`** prefix.

Action

Remove the **`no`** prefix.

---

049 `WARNING: No value given for option [option]. Option ignored.`

Explanation

The named option, appearing in a **`#pragma options xopts`** statement, requires a parameter, but one was not given.

---

050 `WARNING: [option] is not a keyword option. Option ignored.`

Explanation

A parameter was used on the named option in a **`#pragma options xopts`** statement, but the option does not accept a parameter.

---

051 `WARNING: [option] may not be specified via #pragma options. Option ignored.`

Explanation

The named option may be used only when invoking the preprocessor.

Action

Specify the option on the the preprocessor command line or JCL PARM string (on MVS).

---

052 `WARNING: Invalid [option] specification [value]. Option ignored.`

Explanation

In a **`#pragma options xopts`** statement, the value specified for the named option was incorrect or out of range.

---

053 `WARNING: Value of pagesize option, [value], is not numeric or is less than 7. [value] is lines per page.`

Explanation

In a `#pragma options xopts` statement, the value specified for the `pagesize` option is incorrect or out of range. The preprocessor will retain the previously defined `pagesize` value.

Action

Correct the option value. The `pagesize` option requires a numeric parameter that is at least 7.

---

054 `ERROR: Invalid qualification statement. Expected [value/operator] not found.`

Explanation

In a WHERE option, one of the following was expected but not found:

- a field name
- a field value
- a relational operator, such as =, >=, ^=, and so on
- a boolean operator, either AND or OR.

---

055 `ERROR: More than 12 qualification statements specified.`

Explanation

In a WHERE option, more than 12 qualification statements connected by AND or OR were found. No more than 12 qualification statements may be used in a single WHERE option argument.

---

056 `ERROR: More than 12 values specified in FIELDLENGTH.`

Explanation

In a FIELDLENGTH option, more than 12 field length values were found. No more than 12 field length values may be used in a single FIELDLENGTH option argument.

---

057 `WARNING: Field name truncated to [text].`

Explanation

A WHERE option argument contained a field name longer than eight characters. The field name will be truncated to the specified text.

Action

Ensure that the truncated form of the field name is correct.

---

059 **`WARNING: Field name [name] does not begin with alphabetic character`**.

Explanation

A field name must begin with an alphabetic character, but the specified field name does not.

060 **`WARNING: [option] name is longer than [num] characters and will be truncated.`**

Explanation

[option] is either PSB, ID, or SEGMENT. The name, specified as a string literal, was longer than the maximum length permitted by the option. The name will be truncated on the right to the number of characters shown in the message.

062 **`WARNING: USING assumed.`**

Explanation

The word USING was not specified before the PCB option. It is assumed.

Action

Change the command to specify USING PCB.

063 **`WARNING: [option] name is longer than [num] characters and will be truncated.`**

Explanation

[option] is either PSB, ID, or SEGMENT. The name, specified as a string literal, was longer than the maximum length permitted by the option. The name will be truncated on the right to the number of characters shown in the message.

064 **`ERROR: More than 14 parent segments specified. Segment [name] ignored`**.

Explanation

No more than 15 segment names, including the name of the object segment, may be specified in the command.

065 **`WARNING: No options currently stacked. Options are not changed`**.

Explanation

A **`#pragma options pop xopts`** statement was found, but no options have been pushed.

Action

Review the sequence of **`#pragma options push xopts`** and **`#pragma options pop xopts`** statements in the program to determine if the **`pop`** statement is superfluous or a **`push`** statement is missing.

066 **`ERROR: The FORCE option is obsolete. Use FORCEPURGE instead of PURGE FORCE.`**

Explanation

The FORCE option of the SET TERMINAL command is now obsolete.

Action

Re-code the command and replace the PURGE FORCE option with the FORCEPURGE option.

---

067 `NOTE: The @ operator was emitted in certain function calls. Please use the AT option at compile time.`

Explanation

The preprocessor may translate certain uses of the MAPSET option of the SEND MAP and RECEIVE MAP commands using the non-standard "@" operator.

Action

Use the AT compiler option when the preprocessed source code is compiled.

---

068 `ERROR: The EXIT option cannot be used with any of the task-related user exit arguments.`

Explanation

The EXIT option may not be specified in the ENABLE PROGRAM command when any of the FORMATEDF, LINKEDITMODE, SHUTDOWN, TALENGTH, or TASKSTART options is used.

Action

Recode the command without using the EXIT option.

---

069 `ERROR: The default value of the option [option] may be incorrect.`

Explanation

The preprocessor generated a default value for the named option.

Action

Check the generated C program to ensure that the value is correct. If the preprocessor did not choose a correct default value, re-code the command and specify an explicit value for the option.

---

070 `SEVERE: The output filename matches the input filename.`

Explanation

UNIX only: The same filename was specified for both the input and output file. The preprocessor cannot create an output file that would destroy the input file.

Action

Re-run the preprocessor using a different name for the output file.

---

071 `SEVERE: The output filename matches the listing filename.`

Explanation

UNIX only: The same filename was specified for both the output and listing file. The preprocessor can not create both an output file and a listing file with the same name.

Action

Re-run the preprocessor using a different name for one of the files.

---

072 `SEVERE: The listing filename matches the input filename.`

Explanation

UNIX only: The same filename was specified for both the input and listing file. The preprocessor will not create a listing file that would destroy the input file.

Action

Re-run the preprocessor using a different name for the listing file.

---

073 `ERROR: The argument [value] for [option] should not be a string.`

Explanation

A string constant is being passed as an argument to an option that requires a data area (e.g., a 1-value).

Action

Correct the command by replacing the string literal with a 1-value.

---

074 `WARNING: [option] is not valid with the EXCI option, will be ignored.`

Explanation

The named option cannot be used with the EXCI option.

Action

Correct the command before re-invoking the preprocessor.

---

075 `WARNING: [option] is valid only with the EXCI option, will be ignored.`

Explanation

The named option can be used only in conjunction with the EXCI option.

Action

Correct the command before re-invoking the preprocessor.

8 LSCT C++ Translator Messages

This chapter describes diagnostic messages generated by the SAS/C C++ Development System and SAS/C C++ Cross-Platform Development System. Translator diagnostics identify program errors such as missing or misplaced tokens, extraneous tokens, and type mismatches but cannot tell you whether your programming logic is correct.

Each diagnostic includes the message number and text, the error level (for example, ERROR or WARNING), an explanation of the message, and if possible, a solution.

C++ diagnostics have the form

```
LSCT[num] [severity]: [message text]
```

If the translator generates one or more error messages, the source program is incorrect and translation is terminated without producing a compiled output file. Warning messages, on the other hand, indicate potential problems which may not affect how the program is compiled, linked, and executed.

You should fix the errors in the order they appear on your display or listing. Sometimes a single error in your program can result in multiple error messages. The translator attempts to pinpoint the line number in which the error occurred, but occasionally, the error may occur in a line before the line number given by the translator.

Note: Some messages refer you to additional information in the reference section of Bjarne Stroustrup's *The C++ Programming Language, Second Edition* (CPL2).

The C++ message prefix LSCT has been omitted from the descriptions below.

LSCT Messages

101 **`ERROR: Illegal token.`**

Explanation

A symbol that is not recognized as a valid token has been detected in the input. For example, you may have a symbol that you intended to be a C++ identifier, but that contains a character not permitted in identifiers. For instance, symbols cannot have a number as the first character of the identifier.

---

102 **`ERROR: Can't find file: [filename].`**

Explanation

The file specified in a **`#include`** directive cannot be found.

---

103 **`ERROR: Invalid filename.`**

Explanation

The filename in a **`#include`** directive must be enclosed in double quotes (**`" "`**) or angle brackets (**`< >`**).

---

104 **`ERROR: End of file encountered in comment.`**

Explanation

The end of the source file has been detected while a comment was being processed (and before the comment terminator was seen).

105 `WARNING: Invalid escape sequence.`

Explanation

An invalid escape sequence has been detected. An invalid escape sequence is a backslash (\) followed by a character not valid in such a context. For example, \z is not a valid escape sequence.

106 `ERROR: Illegal preprocessor directive.`

Explanation

A # character followed by a symbol not recognized as a valid preprocessor directive has been detected. You may have misspelled the directive.

107 `WARNING: Extra token(s) after preprocessor directive.`

Explanation

A **`#if`**, **`#else`**, or **`#endif`** directive has been followed by some text not within a comment.

108 `ERROR: Missing identifier in preprocessor context.`

Explanation

An identifier is required in the given preprocessor context, but is not present. For example, if a line at your program is **`#if defined`** instead of **`#if defined myfunc`**, this message is issued.

109 `ERROR/WARNING: Redefinition of preprocessor symbol: [symbol].`

Explanation

A previously defined symbol has been redefined. If the definitions are not the same, this is an error.

110 `ERROR: Missing ')' in macro call.`

Explanation

An occurrence of a left parenthesis in a macro call has not been matched by a right parenthesis.

111 `ERROR: Missing argument to preprocessor macro.`

Explanation

A preprocessor macro has been called with fewer arguments than the macro requires. Check the definition of the macro to determine which arguments are missing.

112 `ERROR: Missing comma in preprocessor expression.`

Explanation

A comma is required to separate the arguments in a definition or call of a function-like preprocessor macro.

113 `ERROR: Illegal operator in preprocessor context.`

Explanation

The preprocessor has detected an operator that it cannot understand. Examples of operators that are illegal in the preprocessor context are: ++, - -, . , and ->.

114 `ERROR: Missing operand in preprocessor context.`

Explanation

An operand was expected but not found.

115 `ERROR: Illegal expression in preprocessor context.`

Explanation

The preprocessor has encountered an expression it cannot understand. For example, an operand may be expected but is not present, or the operand is of a non-integral type (for example, a floating-point constant).

116 `ERROR: Preprocessor number not a true number.`

Explanation

A token beginning with a digit resembles an integer or floating-point constant, but is not correct. For example, you may have mistyped a digit in a hexadecimal constant or mistyped a floating-point exponent field.

117 `ERROR: Integer required in preprocessor expression.`

Explanation

A non-integer operand has been detected in a preprocessor expression. The logical operators (!, ||, and &&) require integral operands.

118 **`ERROR: Extra #else or #elif.`**

Explanation

A **`#else`** or **`#elif`** directive has been found without a matching **`#if`** directive.

119 **`ERROR: Extra #endif.`**

Explanation

A #endif directive has been found without a matching **`#if`** directive.

121 **`ERROR: Invalid #line format.`**

Explanation

The preprocessor has encountered an invalid **`#line`** directive. The form of the **`#line`** directive is

```
#line number "string"
```

where **`number`** must be an integer constant and **`''string''`** must be enclosed in double quotes.

122 **`ERROR: Missing parenthesis in preprocessor expression.`**

Explanation

A beginning left parenthesis has been detected in an expression, but it was not matched by a right parenthesis.

123 **`ERROR: Illegal use of # operator.`**

Explanation

The identifier following the # operator is not an argument to the macro being defined.

124 **`ERROR: #error directive.`**

Explanation

The preprocessor has encountered a **`#error`** directive. This message is issued whenever a **`#error`** directive is encountered.

125 **`ERROR: Illegal ## expression.`**

Explanation

Either the first or second operand of the **`##`** operator is missing. This may happen if the **`##`** expression begins or ends the line.

126 `ERROR: Illegal operand of ## operator.`

Explanation

An operand of the `##` expression is not an identifier. You may have misspelled the identifier.

127 `ERROR/WARNING: Unterminated string or character constant.`

Explanation

A string (starting with a double quote) or a character constant (starting with a single quote) was begun but not terminated.

128 `WARNING: No [filename] include file directories specified.`

Explanation

A #include directive was found, but no locations were available for the translator to find header files. See the operating system specific information about running the translator to determine how to specify header file locations.

129 `ERROR: Character literals must contain at least one character.`

Explanation

Two consecutive single quotes were found with no intervening characters. Character literals must have at least one character between the single quotes.

130 `ERROR: Unterminated preprocessor conditional.`

Explanation

The end of the source file was reached while a conditional preprocessor directive (such as the **`#elif`** directive) was pending. One or more **`#endif`** directives are missing.

131 `ERROR: Integer value out of range.`

Explanation

The value of an integer constant or the result of an operation on integers is either too large or too small to be represented. Frequently this is caused by a multiplication or division of two integers.

132 `ERROR: Constant too large for [data-type].`

Explanation

The constant is too large to fit into the named data type.

134 `ERROR: Illegal use of ## operator.`

Explanation

The token created by the `##` operator was not a valid preprocessor token. This may occur, for instance, when an operator token is pasted to an identifier token.

180 `Warning: More than two filename arguments (extras ignored): [filename].`

Explanation

The C++ translator expects two filename options (options without a leading dash)–the first, the input filename, and the second, the output filename. Further filenames are ignored. This message applies only to the SAS/C C++ Cross-Platform Development System.

181 `Warning: Invalid [option-type] option (ignored): [option-name]`.

Explanation

The C++ translator does not recognize this option or sub-option.

182 `WARNING: No such message #[message-number], in option [option-name]. Option (ignored).`

Explanation

The given option contained a message number that is not recognized by the C++ translator.

183 `WARNING: Cannot negate the [option-name] option (ignored).`

Explanation

The given option cannot be negated.

184 `ERROR: Cannot [operation] the [file-type] file named '[filename]'.`

Explanation

The given operation (open, read, or close) could not be performed on the given file.

185 `ERROR: Cannot construct the default [operation] filename from the [operation] filename '[filename]'.`

Explanation

The given operation could not be performed on the given file. (This message applies only to the SAS/C C++ Cross-Platform Development System.)

186 `WARNING: Option [option-name] ignored because it is incompatible with option [option-name].`

Explanation

The first option is incompatible with the previously specified second option.

187 `ERROR: Cannot use output file [filename] because it has a record length of one(1).`

Explanation

Output files must have a record length of at least two (2). (This message applies only to the native 370 SAS/C C++ Development System.)

188 `WARNING: Invalid SNAME: [section-name].`

Explanation

SNAMES must be 1-7 valid C/C++ identifier characters. (This message applies only to the native 370 SAS/C C++ Development System.)

190 `ERROR: Invalid [filename] fileid: [id].`

Explanation

The given filename was either not specified, incorrectly specified, or could not be generated. (This message applies only to the native 370 SAS/C C++ Development System.)

200 `ERROR/WARNING: Syntax error [more-explanation].`

Explanation

A syntax error has been detected. There are many forms of message #200. The amount of information provided in these messages is dependent upon the context in which the error occurs.

Because any **`#pragma`** not recognized by an implementation must be ignored, diagnostics pertaining to **`#pragma map`** and **`#pragma linkage`** are issued as warnings rather than errors. However, you should treat them as errors because no action is taken on a faulty **`#pragma`** of this sort.

205 `ERROR: Newline within string or character literal.`

Explanation

The newline character (indicating an end of line) has been found within a string literal or character literal. Usually this occurs when a terminating single quote or double quote has been omitted from the literal.

206 `ERROR: Bad character in input [hex-number].`

Explanation

The specified character has been detected in the input and is not an acceptable character for a token. Normally this condition occurs only if your source file has had odd characters inserted in it, perhaps as a result of uploading or downloading the file from one machine to another.

207 `WARNING: Asterisk found as first character of C++ style comment.`

Explanation

You may have made an error in starting the comment. Check that it begins as a C style comment (/* */) instead of a C++ style (//) comment.

208 `WARNING: C style comment starting on line line-number never ends.`

Explanation

There is no terminating */ sequence to the comment. Either you have forgotten the comment terminator or you intended the comment to be a C++ style comment.

319 `ERROR: '[identifier]' not declared.`

Explanation

The specified identifier [identifier} is not declared. It may be misspelled.

320 `ERROR: No such class: '[identifier]'.`

Explanation

The specified identifier is not a class, but is used in a place where a class name is required, such as before the `::` operator or in a base-specifier-list.

321 `ERROR: 'struct' or 'class' used on 'enum [identifier]'.`

Explanation

The specified identifier is an `enum` tag, but the keyword `struct` or `class` was used instead of `enum`.

322 `ERROR: 'enum' used on 'class [identifier]'.`

Explanation

The specified identifier is a class tag, but the keyword `enum` was used instead of `struct` or `class`.

323 `ERROR: '[identifier]' previously declared to be a [type-name].`

Explanation

The specified identifier was previously declared to have some other type.

324 `ERROR: '[identifier]' redefined.`

Explanation

The specified identifier was previously defined in this scope.

325 `ERROR: Scoped declaration in parameter list.`

Explanation

The `::` operator cannot be used in parameter lists.

326 `ERROR: Label '[label-name]' not defined.`

Explanation

The specified label appeared as the target of at least one `goto` statement in the previous function, but the label was never defined in the function. Labels are defined in a function using the `label : statement` notation.

327 `ERROR: Label '[label-name]' previously defined.`

Explanation

The specified label was defined more than once in the same function.

328 `ERROR: Repeated keyword or type name: '[keyword]'.`

Explanation

A keyword or type name was used more than once within a single declaration.

329 `ERROR: Conflicting keywords or type names: '[keyword-1]' and '[keyword-2]'.`

Explanation

An illegal combination of keywords or type names was used in this declaration.

330 `ERROR: Must be integral, pointer, or member pointer.`

Explanation

An expression of a non-testable type was used where a testable value is required. Testable types are all the integral, pointer and member-pointer types. Testable values are required as the first expression of a `?:` operator and as the test for the `if, while, do while,` and `for` statements.

331 `ERROR: Must be integral.`

Explanation

An expression of a non-integral type was used where an integral valued expression is required. An integral value is required in a `case` label and in the test expression of a `switch` statement.

332 `ERROR: No such conversion: to ([type]) from ([type] ).`

Explanation

An illegal conversion was specified in a cast operator.

334 `ERROR: Expression is not modifiable.`

Explanation

You have tried to assign, increment, or decrement something that was not a modifiable lvalue.

335 `ERROR: Invalid use of '&' address-of operator. [object].`

Explanation

The address-of operator (&) was applied to an object that was not addressable. Examples of nonaddressable objects are bitfields and register variables. Also, you cannot take the address of an overloaded function except as an initializer.

336 `ERROR: Cannot initialize [variable] with [initializer].`

Explanation

The initializer is of a type that cannot be converted to the type of the variable it is initializing.

337 `ERROR: Preprocessor error.`

Explanation

The preprocessor has encountered an error that is beyond its capabilities to diagnose.

Action

Contact SAS/C Technical Support.

338 `ERROR: Unexpected end of file.`

Explanation

The parser reached the end of the input source before it expected to. This may occur during error recovery from a previous syntax error in the input source. Otherwise, it usually indicates that a closing brace (}) or semicolon (;) has been omitted at the end of your source.

339 `WARNING: A non-lvalue array was converted to a pointer.`

Explanation

Only arrays that are lvalues can be converted to pointers. Because the array is not converted to a pointer, `operator []` should not be applied to it because `operator []` requires a pointer. Also the array should not be assigned to a pointer variable. If you want only to access the array, you may ignore this warning. However, if you want to alter the value of an array element, you should treat this message as an error.

340 `ERROR: The base name '[class-1]' is ambiguous in class '[class-2]'.`

Explanation

The class [class-1] occurs more than once as a base of the second class, [class-2].

342 `ERROR: Conversion from a virtual base class '[class-name]' to a derived class is not allowed.`

Explanation

Virtual base classes cannot be converted, either explicitly or implicitly, to derived classes.

343 `ERROR: Ambiguous conversion to integral type from 'class [class-name]'.`

Explanation

The class has defined multiple conversions to an integral type.

344 `ERROR: Ambiguous conversion to pointer from 'class [class-name]'.`

Explanation

The class has defined multiple conversions to pointer.

345 `ERROR: Ambiguous conversion to testable from 'class [class-name]'.`

Explanation

The class has defined multiple conversions to one or more of arithmetic, pointer, or member-pointer types.

346 `ERROR: Ambiguous conversion to derived member pointer.`

Explanation

An ambiguous reference to the derived member-pointer has been found. Resolve the ambiguity by qualifying the pointer name with its class name.

347 `ERROR: Ambiguous conversion of overloaded function pointer.`

Explanation

An ambiguous reference to the overloaded function pointer has been found. Resolve the ambiguity by qualifying the function pointer name with its class name.

348 `ERROR: Ambiguous conversion to class.`

Explanation

Compiler found an ambiguous reference to the conversion to a class. For example, you may have a file-scope function that has the same name as a class (hence, the same name as the conversion function of that class).

349 `ERROR: Ambiguous conversion.`

Explanation

An ambiguous conversion has been specified. For example, a constructor and a function cannot have the same name.

350 `ERROR: Ambiguous function call.`

Explanation

The function name used in the call is ambiguous. For example, two classes using different functions with the same name may not have used a class name to qualify the function name called.

351 `ERROR: Overloaded functions '[function-1]' and '[function-2]' used ambiguously in conditional expression.`

Explanation

Two overloaded functions were used as the second and third operands to a conditional operator (`?:`). These overloaded functions have more than one function type in common. Cast one or both operands to the desired function pointer or member function pointer type.

352 `ERROR: Ambiguous common base class: [class-name].`

Explanation

The reference to the base class is ambiguous. Resolve the ambiguity by further qualifying each occurrence of this name.

353 `ERROR: Ambiguous member name: [member-name].`

Explanation

The expression used to refer to the member could refer to more than one function, object, type, or enumerator. Resolve the ambiguity by qualifying the member name with its class name.

354 **`ERROR: Non-static member '[member-name]' must be used with dot, arrow, or address-of operator.`**

Explanation

Non-**`static`** members can be used only in the following contexts:

- after a dot or arrow operator
- as a member-pointer (as in **`&class-name::member-name`**)
- within a **`sizeof`** or **`offsetof`** expression
- within a non-**`static`** member function of a class that contains or inherits the member (where the **`this->`** operation is implied).

355 **`ERROR: Value of an undefined class cannot be used.`**

Explanation

You have tried to dereference a pointer to an undefined class. A solution is to include the definition of the class before it is used.

356 **`ERROR: An array may not be the target of an assignment.`**

Explanation

Array assignment (that is, assignment of an array name) is not supported in C or C++.

357 **`ERROR: A function may not be the target of an assignment.`**

Explanation

Function assignment (that is, assignment to a function name) is not supported in C or C++.

358 **`ERROR: Cannot [operation] a pointer to [type].`**

Explanation

You cannot add to or subtract from a pointer to **`void`**, a function pointer, or a pointer to an undefined class. Nor can you use the indirection operator (*) on a pointer to **`void`** or a pointer to an undefined class. The error message explains which of these mistakes you have made.

359 **`ERROR: Typedef names cannot be declared in parameter lists.`**

Explanation

A **`typedef`** name has been encountered in a parameter list. Move the definition of the **`typedef`** name outside the parameter list (that is, to file scope).

361 `ERROR: Cannot take the address of a member of virtual base class.`

Explanation

The & operator cannot be applied to a member of a virtual base class.

362 `ERROR: Invalid initializer.`

Explanation

The initializer is of a type that cannot be converted to a type required by the variable it is initializing.

363 `ERROR: Invalid use of void.`

Explanation

Only functions or pointers can be declared **`void`**.

364 `ERROR: Cast to undefined class not allowed.`

Explanation

The class to which a cast is made must be previously defined. Also, for **`dynamic_cast`**, when casting to a class pointer, the class type must be previously defined.

365 `ERROR: Cannot find offset into non-class.`

Explanation

Only members of classes can have offsets.

366 `ERROR: Cannot find offset into undefined class.`

Explanation

A class must be defined before the offset of one of its members can be taken.

367 `ERROR: Invalid use of the scope operator.`

Explanation

The scope operator **`::`** can be used only in such expressions as **`C3::mem`** or **`C1::C2::C3::mem`**, where **`C1`** is a class in which class **`C2`** is declared, **`C3`** is a class declared in **`C2`**, and **`mem`** is a member of **`C3`**.

368 `ERROR: Cannot find the offset of 'object'.`

Explanation

It is illegal to take the offset of a member function, a static member or a bitfield member. In particular, because the number of bits in a bitfield may be less than the number of bits in a **`char`**, and its number of bits may not comprise an integral number of **`chars`**, a bitfield in C++ has no size.

369 `ERROR: Cannot find offset because class '[class-name]' has no member named '[member-name]'.`

Explanation

The second argument to the offset operation must be a member of the class specified in the first argument.

370 `ERROR: Cannot take the size of an undefined class.`

Explanation

A class must be defined before its size can be determined.

371 `ERROR: Cannot dereference pointer to undefined class.`

Explanation

A class must be defined before a pointer to an object of its type can be dereferenced.

372 `ERROR: No such constructor.`

Explanation

The constructor referred to does not exist.

373 `ERROR: '[identifier]' previously declared as [type-1]. Cannot be defined as [type-2].`

Explanation

A name declared using the **`union`** specifier cannot be defined using **`struct`** or **`class`**. Similarly, a name declared as **`struct`** or **`class`** cannot be defined using **`union`**.

374 `ERROR: No such member '[member-name]'.`

Explanation

The identifier referred to is not a class member.

375 `ERROR: Member '[member-name]' redeclared.`

Explanation

The specified member of the class has been declared more than once.

376 `ERROR: '[identifier]' not a definable member.`

Explanation

Only functions and static data members can be defined outside the class declaration.

377 `ERROR: 'this' may occur only in a (non-static) member function.`

Explanation

You cannot use the **this** keyword outside the context of a non-**static** member function.

378 `ERROR: Cannot create a new value of a function.`

Explanation

The **new** operator cannot be applied to a function type. Functions cannot be allocated by means of the **new** operator.

379 `ERROR: Cannot create a new value of a reference.`

Explanation

The **new** operator cannot be applied to a reference type. Because a reference type is not an object, a pointer to it could not be returned by **operator new**.

380 `ERROR: Cannot create a new instance of an undefined class.`

Explanation

A class must be fully defined before the **new** operator can be applied to it to create a new instance of the class.

382 `ERROR: Missing array size in expression.`

Explanation

One or more dimensions of the given array have not been specified.

383 `ERROR: Class '[class-name]' has no default constructor.`

Explanation

You cannot use **operator new** to allocate an array of class objects if the class does not have a default constructor.

384 `ERROR: Cannot initialize new array.`

Explanation

An array created by means of the **new** operator cannot be initialized by specifying a brace-enclosed initializer list.

385 `ERROR: Cannot delete an object of an undefined class.`

Explanation

The **delete** operator cannot be applied to an object whose class has not been defined.

386 `ERROR: Length expression of array must be integral.`

Explanation

The size of an array cannot be specified as a floating-point number.

387 `ERROR: No match for call to function or overloaded operator.`

Explanation

The argument list given for a function call did not match any of the possible parameter lists.

388 `ERROR: Missing constructor body.`

Explanation

A constructor declaration was followed by a colon (:), but no constructor body ({) was found after that.

389 `ERROR: Non-virtual functions '[function-name]' cannot be declared pure.`

Explanation

Only virtual functions can be declared pure.

391 `ERROR: Uninitialized const identifier.`

Explanation

A const identifier must have an explicit initialization.

392 `ERROR: Uninitialized const identifier or reference: [identifier].`

Explanation

A const identifier must be initialized explicitly. The declaration of a reference must contain an explicit initializer unless one of the following is true:

- the **`extern`** specifier has been used
- the reference declaration is a class member declaration within a class declaration
- the reference declaration is a declaration of a function parameter or function return type.

393 `ERROR: Const identifier or reference member '[member-name]' must be initialized.`

Explanation

A **`const`** identifier or reference must be initialized explicitly. This message is caused by a constructor for a class with a **`const`** or reference member where the **`const`** or reference member is not initialized with a mem-initializer in the constructor. This message is also output for aggregate initialization of a structure that does not initialize a reference member.

394 **ERROR: Member '[member-name]' must have initializer, class '[class-name]' has no default constructor.**

Explanation

If a class has a constructor (but does not have a default constructor), objects of that class must be initialized. A member is initialized by including a mem-initializer for it on each constructor for the class containing the member.

For example, if the type of **X::a** is a class with a constructor (but no default constructor) you receive this message if you omit the **a(10)** in the following code:

```
X::X() : a(10), b(11)

{ . . . }
```

395 **ERROR: Base '[class-name]' must have initializer, class '[class-name]' has no default constructor.**

Explanation

If a class has a constructor (but does not have a default constructor), it must be initialized. A base class is initialized by including a mem-initializer for it on each constructor for the derived class.

For example, if **b** is a base class of **X** and is a class with a constructor (but no default constructor), you receive this message if you omit the **b(11)** in the following code:

```
X::X() : a(10), b(11)

{ . . . }
```

396 **ERROR: Virtual base class '[class-name]' must have initializer since class has no default constructor.**

Explanation

If a class has a constructor (but does not have a default constructor) it must be initialized. A virtual base class is initialized by including a mem-initializer for it on each constructor for the derived class. The example in message #395 is applicable in the virtual base class case as well.

397 **ERROR: '[identifier]' is not a direct base class or member of class '[class-name]'.**

Explanation

The specified identifier is not a member or base class of the class being constructed.

398 `ERROR: Member access through protected base class not allowed for '[member-name]'.`

Explanation

The specified member cannot be accessed because it has been inherited from a protected base class and the function or initializer using the member is not a friend or member of a class derived from that base.

399 `ERROR: Member access through private base class not allowed for '[member-name]'.`

Explanation

The specified member cannot be accessed because it has been inherited from a private base class and the function or initializer using the member is not a friend or member of the class that is derived directly from that base.

400 `ERROR: Base access through protected base class not allowed.`

Explanation

The expression that is attempting to access the base class is not in a function or initializer that has access to it. Because the base class is protected, only functions of the following types have access to the base class:

- members or friends of the class that declared the base class
- members or friends of classes derived from the class declaring the base class.

Initializers for members of a class have the same access privileges as functions of that class.

401 `ERROR: Base access through private base class not allowed.`

Explanation

The expression that is attempting to access the base class is not in a function or initializer that has access to it. Because the base class is private, only functions that are members or friends of the class that declared the base class have access to the base class. Initializers for members of a class have the same access privileges as functions of that class.

402 `ERROR: Cannot access protected member '[member-name]'.`

Explanation

The expression that is attempting to access the member **`member-name`** is not in a function or initializer that has access to it. Because **`member-name`** is protected, only functions of the following types have access to **`member-name`**:

- members or friends of the class that declared **`member-name`**
- members or friends of classes derived from the class declaring **`member-name`**.

Initializers for members of a class have the same access privileges as functions of that class.

403 `ERROR: Cannot access private member '[member-name]'.`

Explanation

The expression that is attempting to access the member **member-name** is not in a function or initializer that has access to it. Because **member-name** is private, only functions that are members or friends of the class that declared **member-name** have access to **member-name**. Initializers for members of a class have the same access privileges as functions of that class.

404 `ERROR: Virtual function '[function-name]' declared in virtual base '[class-name]' must be overridden.`

Explanation

The virtual function is declared in a virtual base class. It is also overridden in at least two classes derived from the base class and inherited by the class that caused this message. The specified virtual function must be declared in the class that caused this message.

406 `ERROR: Parameter of type 'void'.`

Explanation

Parameters to functions cannot be declared to be of type **void**.

407 `ERROR: Default argument expression missing.`

Explanation

If a formal parameter has a default argument value, then all the parameters after this one must also have default argument values.

408 `ERROR: Multiple declarations of function specifying default arguments.`

Explanation

The default argument for a formal parameter can be given in only one function declaration.

409 `ERROR: Arrays cannot contain elements of type 'void'.`

Explanation

Array elements must be of some type other than **void**.

410 `ERROR: Arrays cannot contain bitfields.`

Explanation

Array elements must be of some type other than bitfield.

411 `ERROR: Arrays cannot contain functions.`

Explanation

You can define an array of function pointers, but not an array of functions.

412 `ERROR: Functions cannot return functions.`

Explanation

A function can return a function pointer, but it cannot return a function.

413 `ERROR: Functions cannot return arrays.`

Explanation

A function can return a pointer, but it cannot return an array.

414 `ERROR: Functions cannot return bitfields.`

Explanation

Bitfields cannot be returned by functions.

415 `ERROR: Functions cannot return undefined classes.`

Explanation

If a function is intended to return a class, that class must first be defined.

416 `ERROR: Pointers cannot point to references.`

Explanation

References are not addressable and so cannot be referenced by pointers.

417 `ERROR: Pointers cannot point to bitfields.`

Explanation

Bitfields are not addressable and so cannot be referenced by pointers.

418 `ERROR: References cannot refer to references.`

Explanation

A reference whose value is a reference is not permitted.

419 `ERROR: References cannot refer to bitfields.`

Explanation

A reference cannot refer to a bitfield.

420 `ERROR: References cannot refer to objects of type 'void'.`

Explanation

There are no **void** objects, so there cannot be a reference to one.

421 `ERROR: Member pointers cannot point to bitfields.`

Explanation

Bitfields cannot be referenced by member-pointers.

422 `ERROR: Member pointers cannot point to references.`

Explanation

A reference cannot be referred to by a member-pointer.

423 `ERROR: Member pointers cannot point to objects of type 'void'.`

Explanation

Because there are no **void** objects, a pointer cannot point to one.

424 `ERROR: Bitfields must be of integral type.`

Explanation

Bitfields cannot be of floating-point type.

425 `ERROR: Overloaded functions with indistinguishable arguments.`

Explanation

Two functions have the same name and the same parameter list. Either delete one of the functions, or change its definition.

426 `WARNING: K&R C style function definition.`

Explanation

A Kernighan and Ritchie (K&R) C style function definition has been encountered. C++ requires function definitions to be of prototypical form, but K&R style function definitions are permitted as an extension. This message is only to warn you about the use of this extension. If you do not want to see this warning, use the **suppress** translator option to turn it off.

427 `ERROR: K&R C style functions cannot return classes with constructors or destructors.`

Explanation

If a function is to return a class with a constructor and destructor, it must be defined by means of a prototypical function definition, as opposed to using a Kernighan and Ritchie (K&R) C style function definition.

428 `ERROR: Conversion function must be a member function.`

Explanation

A conversion function for a class must be defined as a member function.

429 `ERROR: Destructor function must be a member function.`

Explanation

The destructor for a class must be defined as a member function.

430 `ERROR: Conversion function '[function-name]' not correctly declared.`

Explanation

A return type was specified for a conversion function, or formal parameters were given for a conversion function. Declarations of conversion functions do not specify the return type in the usual way. The return type is part of the name of the function. Also, conversion functions cannot take arguments.

431 `ERROR: Destructor function '[destructor]' not correctly declared.`

Explanation

A destructor has been declared to be something other than a function, or a return type was specified for a destructor. Destructors must be functions and declarations of destructors cannot specify a return type (not even `void`).

432 `ERROR: Copy constructor for a class may not take an argument whose type is that class.`

Explanation

Copy constructors cannot take an argument whose type is the class of which the copy constructor is a member. Typically, you can copy objects of `class ABC` by declaring a copy constructor of the form `ABC::ABC(const ABC&)`.

433 `ERROR: Operator function '[function-name]' not correctly declared.`

Explanation

An operator function must be either a member function or have at least one parameter of type class.

434 `ERROR: Invalid linkage specifier.`

Explanation

The **extern** keyword must be followed by a string literal containing either `''C''` or `''C++''`. This error may also be caused by an extra or misspelled token after an extern keyword. `''C''` and `''C++''` must be specified in uppercase.

435 `ERROR: Linkage differs from prior declaration.`

Explanation

The linkage specified is not the same as that specified in a prior declaration of the function.

436 `ERROR: Unknown linkage convention.`

Explanation

SAS/C C++ implements only the `''C''` and `''C++''` linkage conventions. Linkage to other languages must be specified using the SAS/C language keywords. For more information on using other languages in combination with C, see the *SAS/C Compiler Interlanguage Communication Feature User's Guide.*

437 `ERROR: Missing class name.`

Explanation

A class name was expected but not found. This may be caused by a misplaced comma.

438 `ERROR: Repeated base class.`

Explanation

A base class can be mentioned only once in the base-list.

439 `ERROR: Objects of abstract classes '[object-name]' cannot be declared.`

Explanation

Classes that have pure virtual functions (abstract classes) can be used only as base classes; they cannot be used to declare variables or members of other classes. It is permissible to declare pointers to abstract classes.

440 `ERROR: Object of type 'void'.`

Explanation

Only a function or a pointer can be declared to be of type `void`.

441 `ERROR: Static members '[member-name]' of a local class may not be initialized.`

Explanation

This is one of the limitations of local classes. All `static` data members of local classes are automatically initialized to zero.

442 `ERROR: Cannot use undefined enum '[identifier]'.`

Explanation

Enumerations must be defined before they can be used.

443 `ERROR: Enum constants '[identifier]' must be initialized with integral values.`

Explanation

The values of an enumeration must be integral values (rather than, for example, floating-point values).

444 `ERROR: A class cannot be a member of itself.`

Explanation

No member of a class can be of the same type as that class (although a class can have as a member a pointer or reference to that type of class).

445 `ERROR: Cannot declare members of an undefined class.`

Explanation

A class must be defined before any of its members can be declared.

446 `ERROR: Cannot declare arrays of an undefined class.`

Explanation

If an array of instances of a given class is to be declared, the class must first be defined.

447 `ERROR: Cannot declare variables of an undefined class.`

Explanation

Classes must be defined before objects of that class can be declared.

448 `ERROR: Cannot initialize data members in member declaration.`

Explanation

Data members cannot be initialized within the class definition. Non-`static` data members must be initialized in the mem-initializer of each constructor function. `Static` data members must be initialized outside the class.

449 `ERROR: Member function of a local class must be defined within that class: [class-name].`

Explanation

A member function of a local class must be defined within that class and not merely declared within the class.

450 `ERROR: Member '[member-name]' declared 'void'.`

Explanation

A member cannot be of type `void`.

451 `ERROR: 'friend' used on non-function.`

Explanation

The `friend` keyword has meaning only in function declarations inside a class.

452 `ERROR: 'friend' can only be used inside a class.`

Explanation

The `friend` keyword has meaning only in function declarations inside a class.

453 `ERROR: Invalid syntax for access declaration.`

Explanation

A member declaration contained inconsistent information such as:

- the friend function declaration omitted the required `friend` keyword
- the access declaration specified type information
- the member declaration inside the class definition specified the class using the scope operator.

Either add the `friend` keyword, remove the type information, or remove the class name and scope operator, depending upon which type of declaration you intended.

454 `ERROR: Invalid access adjustment: '[member-name]'.`

Explanation

Access to a base class member cannot be adjusted in a derived class that defines a member of the same name.

455 `ERROR: Access cannot be changed, but only reinstated.`

Explanation

Access declarations must declare the inherited member to have the same access as the member in the class from which it is inherited.

456 `ERROR: Previously declared as a member in this class.`

Explanation

An access declaration cannot specify a name defined in the derived class.

457 `ERROR: '[class::member]' is not a member of a base class.`

Explanation

The specified member is not a member of any base class.

458 `ERROR: Access declaration names class that is not a base of this class.`

Explanation

You have tried to control access to a member of a base class by using an access declaration, but the base class name you have used is not truly a base class of the derived class. You may have misspelled the base class name.

460 `ERROR: Constructor function '[constructor]' not correctly declared.`

Explanation

An incorrectly declared constructor has been found. Perhaps you have specified a return type for a constructor, or the name used in the declaration of the constructor is not the same as the name of the class. Declarations of constructors cannot specify a return type (not even `void`), and the name of the constructor must be the same as the class name.

461 `ERROR: Destructor function '[destructor]' not correctly declared.`

Explanation

An incorrectly declared destructor has been found. Perhaps you have declared it to be something other than a function or specified a return type for the destructor. Destructors must be functions, and declarations of destructors cannot specify a return type (not even `void`). The name of the destructor must be a tilde (~) followed by the class name.

462 `ERROR: Operator function '[function-name]' not correctly declared.`

Explanation

An incorrectly declared operator function has been found. Perhaps you declared it with the wrong number of arguments. Unary operators take only one argument and binary operators take two arguments. Also, member functions have an implicit **`this`** argument, which counts against this limit. So, for example, a unary operator declared as a member function has no explicit formal parameters.

463 `ERROR: Static functions '[function-name]' cannot be virtual.`

Explanation

`static` functions cannot be virtual. Remove the **`virtual`** keyword from the declaration of the **`static`** function.

464 `ERROR: Constructors '[constructor]' cannot be virtual.`

Explanation

Constructor functions cannot be virtual. Remove the virtual keyword from the declaration of the constructor function.

465 `ERROR: Static functions '[function-name]' cannot be used to override virtual functions.`

Explanation

A **`static`** member function was declared to have the same name and argument types as a virtual function inherited from a base class. Use a different name for the **`static`** function.

467 `ERROR: Linkage specification cannot be used in a member declaration '[member-name]'.`

Explanation

Linkage declarations can be used only in file-scope declarations of non-members.

468 `ERROR: Cannot define classes or enums in return types or parameter lists.`

Explanation

Classes and enumerations cannot be defined in parameter lists (prototypes) or in return type declarations.

469 `ERROR: Invalid parameter name '[parameter]'.`

Explanation

Parameter names cannot be operator function names, operator conversion names, or destructor names.

472 `ERROR: Formal '[argument]' is not listed in function declaration.`

Explanation

All arguments to a function must be listed in the parameter list of the function.

473 `ERROR: Initialized local extern '[variable]'.`

Explanation

External variables declared inside a function cannot be initialized in the function. Instead, they must be initialized by another declaration outside the function.

474 `ERROR: A typedef name ('[name]') cannot be initialized.`

Explanation

Declarations of **typedef** names cannot contain initializers.

475 `ERROR: Class with constructors must have an initializer.`

Explanation

Variables declared to be of classes that have constructors must be initialized.

476 `ERROR: Cannot define classes or enums in type names.`

Explanation

Classes and enumerations cannot be defined in cast operators, **new** operators, **sizeof** expressions, or **offsetof** expressions.

477 `ERROR: Not a function.`

Explanation

A file-scope declaration followed by a mem-initializer or a function body must declare a function.

478 `ERROR: A mem-initializer may be used only within constructor functions.`

Explanation

This message occurs when a colon follows a function declarator, but the function is not a constructor.

479 **`ERROR: Base or member '[identifier]' re-initialized.`**

Explanation

The same base class or member was initialized by two mem-initializers in the same constructor function. For example, the following code generates this message because the **`a(10)`** part appears twice:

```
X::X() : a(10), a(10)

{ . . . }
```

---

480 **`ERROR: Old style base initializer cannot be used on class with no bases.`**

Explanation

An old style base initializer is a mem-initializer with no specified name. They can be used only for classes with a single base class. For example, if **`X`** has no base classes, the following code tries to initialize a non-existent base and generates this message:

```
X:: X() : (10)

{ . . . }
```

To correct the error, delete the **`(10)`** part.

---

481 **`ERROR: Old style base initializer cannot be used on class with multiple base classes.`**

Explanation

An old style base initializer is a mem-initializer with no specified name. They can be used only for classes with a single base class. For example, if **`X`** has more than one base, the following code is ambiguous and generates this message:

```
X::X() : (10)

{ . . . }
```

To correct the error, insert the name of a specific base class before the **`(10)`**.

---

482 **`WARNING: Statement is unreachable.`**

Explanation

The statement flagged will never be executed. You may want to check your program logic.

---

483 `ERROR: 'case' label must be within a switch statement.`

Explanation

A **case** label is not allowed outside of a **switch** statement.

484 `ERROR: 'default' label must be within a switch statement.`

Explanation

The **default** label is not allowed outside of a **switch** statement.

485 `ERROR: 'continue' must be within a loop ('do', 'for', or 'while') statement.`

Explanation

The **continue** statement is not allowed outside of a loop statement.

486 `ERROR: 'break' must be within a switch or loop ('do', 'for', or 'while') statement.`

Explanation

The **break** statement is valid only within **switch** or **loop** statements.

487 `ERROR: Missing return value.`

Explanation

The return value of the function was not specified.

489 `ERROR: Return value given for constructor, destructor, or void function.`

Explanation

A return value is not allowed in a constructor, destructor, or **void** function.

490 `ERROR: Missing function name in function declaration.`

Explanation

A function name was expected but not found. You may have misspelled the name of a constructor function.

491 `ERROR: Illegal formal declaration list in prototype function definition.`

Explanation

A function definition included both a prototype and one or more K&R C style argument declarations. These two styles cannot be mixed in a single definition.

492 `ERROR: Formal '[argument]' must be declared in function header identifier list.`

Explanation

In K&R C style function definitions, formals declared in the formal declaration list must have already been specified in the identifier list of the function.

493 `ERROR: Expression in array declarator must be constant expressions.`

Explanation

Array declarations that specify a size must specify it as a non-negative integral constant expression.

494 `ERROR: Expression in array declarator must be integral.`

Explanation

Array declarations that specify a size must specify it as a non-negative integral constant expression.

495 `ERROR: Expression in array declarator must be positive.`

Explanation

Array declarations that specify a size must specify it as a non-negative integral constant expression.

496 `ERROR: Map directive does not match prior directive.`

Explanation

The map directive specified is not the same as that specified in a prior declaration of the symbol.

498 `ERROR: Invalid bitfield size.`

Explanation

Bitfield sizes must be a non-negative constant expression less than or equal to the size of the specified allocation unit.

499 `ERROR: Cannot use undefined class '[class-name]' as base class.`

Explanation

The specified class must be defined before it can be used as a base class for another class.

500 `ERROR: Missing declaration-specifier.`

Explanation

Declaration specifiers are required in all non-function declarations. They are also required in function declarations in parameter lists.

501 `ERROR: Illegal use of '[item]' in local member function.`

Explanation

Local member functions can use type names, **`static`** variables, **`extern`** variables and functions, and enumeration constants only from the enclosing scope.

502 `ERROR: A class cannot be derived from a union ['union-name'].`

Explanation

A class cannot be derived from a union.

503 `ERROR: A union ['union-name'] cannot be derived from another class.`

Explanation

A union cannot be derived from another class.

504 `ERROR: Constant expression contains a division by zero (0).`

Explanation

Constant expressions cannot contain division by zero.

506 `ERROR: Cannot take the size of a function.`

Explanation

Functions have no size, although function pointers do.

507 `ERROR: Cannot take the size of a bitfield.`

Explanation

Bitfield sizes are not expressible in bytes.

508 `ERROR: Cannot take the size of void.`

Explanation

Because there are no `void` objects, you cannot take the size of one.

509 `ERROR: Cannot take the size of array with unspecified length.`

Explanation

The array was declared without giving its length, so its size cannot be determined.

510 `WARNING: Cannot jump into a block to a label after a declaration having an initializer.`

Explanation

You can't use a `goto` statement to jump into a block if the destination label occurs after a declaration has been initialized. Should such a jump occur, the object in question would not be initialized but could be referenced in subsequent code.

511 `ERROR: Overloaded member functions '[function-name]' may not be both static and non-static.`

Explanation

All member functions of the same name must be either `static` or non-`static`.

512 `ERROR: Function hides a virtual function from base class.`

Explanation

Because the function hides a virtual function, the virtual function is not called.

513 `ERROR: Overriding virtual function '[function-name]' has different return type.`

Explanation

An overriding virtual function cannot change the return type.

514 `ERROR: Arrays cannot contain references.`

Explanation

Arrays of references are not allowed.

515 `ERROR: Previous declaration of function had different return type.`

Explanation

An earlier declaration of the function specified a different return type.

516 `ERROR: Cannot have two extern ''C'' functions with same name '[name]'.`

Explanation

In a program, only one of a set of overloaded functions of a given name can be declared `extern ''C''`.

517 `ERROR: Previous declaration differed in the use of __builtin.`

Explanation

All declarations of a function must be consistent in the use of the `__builtin` keyword. All must have it, or none must have it.

518 `ERROR: [object-type] '[expression]' cannot be used in default argument expressions.`

Explanation

Non-`static` members, formal parameters, and automatic variables cannot be used in default argument expressions.

521 `ERROR: Ambiguous use of keyword.`

Explanation

An ILC language keyword was used in a place where it could apply to two different functions. Follow these rules when placing ILC language keywords:

- language keywords applied to functions should be placed directly before the name of that function.
- language keywords applied to function pointers should be placed immediately before the parenthesis that precedes the asterisk representing that pointer.

522 `ERROR: Keyword can only be used on functions.`

Explanation

A keyword that can be used only in function declarations has been used in a declaration of some other type. For example, `virtual` and `inline` can be used only in function declarations.

523 `ERROR/WARNING: '[keyword]' cannot be applied to [object-type].`

Explanation

It is invalid to apply the keyword in the specified context. For example, it is meaningless to apply a storage class keyword such as **`auto`** to the definition of a class (although **`auto`** can be applied to the definition of an object whose type is that class). Depending upon the combination of keywords in question, this may be treated as a warning or an error.

524 `ERROR: Previous declaration was not static.`

Explanation

A function was first declared non-**`static`** and later declared **`static`**.

525 `ERROR: Function declared 'inline' after first use.`

Explanation

A function was used prior to its declaration as an inline function.

526 `ERROR: Prototypes can only be specified for C, C++, and __asm functions.`

Explanation

Prototypes for functions in other languages are not supported.

527 `ERROR: Only C, C++, and __asm functions can be overloaded.`

Explanation

Functions written in other languages cannot be overloaded.

528 `ERROR: Member functions must be C++ functions.`

Explanation

Only C++ functions (and not functions in C or in other languages) can occur as member functions.

529 `ERROR: Keyword can only be used on function pointers.`

Explanation

The keywords **`__local`** and **`__remote`** can be used only on function pointers.

531 `ERROR: A declaration must declare something.`

Explanation

An empty declaration has been encountered. This message usually is caused by the omission of a variable name in a declaration (leaving only a sequence of keywords or type symbols).

532 `ERROR: [function-name] cannot have '[storage-type]' storage class.`

Explanation

A member function declared `const` or `volatile` cannot also be declared `static`.

533 `WARNING: Extra comma at end of enumeration list.`

Explanation

An enumeration list has an extraneous comma after its last member.

534 `WARNING: Enum value: [value] is used for both '[enum-1]' and '[enum-2]'.`

Explanation

An enumeration value has been repeated. This warning is given if two `enum` constants in the same enumeration type have the same value. This may be what you intended; the warning is given in case it is not intended.

535 `ERROR: Cannot overload 'main' or '_dynamn'.`

Explanation

Neither `main` nor `_dynamn` can be overloaded.

536 `ERROR: Cannot call or take the address of 'main'.`

Explanation

The function `main` cannot be called, nor can its address be taken.

537 `ERROR: 'main' cannot be 'storage-type'.`

Explanation

The function 'main' cannot be declared `static` or `inline`.

538 `ERROR: Anonymous classes cannot have constructors or destructors.`

Explanation

An anonymous (unnamed) class cannot have a constructor or a destructor.

539 `ERROR: Destructor names '[destructor]' must be the same as their class names '[class]'.`

Explanation

The tilde (~) can be used only to declare a destructor if that destructor has the same name as the class for which it is a destructor.

540 `ERROR: Expression in array declarator must not be negative.`

Explanation

In C++, an array must be declared to be of a positive size. As an extension, SAS/C C++ allows the declaration of zero-length arrays.

541 `ERROR: Cannot allocate array of class '[class-name]' with no default constructor.`

Explanation

In order to allocate an array of a class with the **new** operator, the class must have a default constructor.

542 `ERROR: Invalid constructor given for member '[member-name]'.`

Explanation

An invalid constructor has been encountered. This happens in two circumstances:

- in constructors that do not give an explicit member initializer for a member that is an array of classes without a default constructor
- in constructors that give a multi-argument member initializer for a non-class member.

543 `ERROR: '[operand-1]' and '[operand-2]' are not compatible types for conditional operator.`

Explanation

The two operands are not of compatible type for use with the conditional operator.

544 `ERROR: [type-1] [operator] [type-2]: Invalid type for binary operator.`

Explanation

One of the types displayed is inappropriate for the operator in question or is incompatible with the other type in this context.

545 `ERROR: '[operand]' is of invalid type for postfix operator '[operator]'.`

Explanation

The postfix operator (++ or —) cannot be applied to an object of the given type.

546 `ERROR/WARNING: '[operator]' is invalid for operand type '[operand]'.`

Explanation

The operator cannot be applied to an operand of this type. Depending upon the operator/operand pair in question, this may be either a warning or an error.

547 `ERROR: '[object]' is of invalid type for call operator.`

Explanation

The call operator (()) cannot be applied to an object of the given type.

548 `ERROR: Invalid pointer conversion from '[type-1]' to '[type-2'].`

Explanation

It is not possible to convert a pointer to an object of the given type.

549 `WARNING: Non-const and/or non-volatile member function called with const and/or volatile object.`

Explanation

It is an error to call a non-**`const`** member function for a **`const`** object or a non-**`volatile`** member function for a **`-volatile`** object, but because many other compilers fail to diagnose this error, SAS/C C++ treats it as a warning. Ignoring this warning allows non-**`const`** functions to change data declared as **`const`**.

550 `WARNING: Non-constant reference ([reference]) bound to a non-lvalue.`

Explanation

The reference in question is to a non-**`const`** but has been initialized with something that is not an lvalue.

551 `ERROR: Cannot take size of pointer to overloaded function '[function-name]'.`

Explanation

Because an overloaded function name does not uniquely determine the function designated, a pointer to such a function likewise is not uniquely determined and consequently its size may be indeterminate as well. Accordingly, **`sizeof`** cannot be applied to such a pointer.

552 `WARNING: Address of temporary taken by casting to reference, ([reference]).`

Explanation

Care should be taken with the result of such a cast because the resulting reference may outlive the temporary it refers to, resulting in a dangling reference.

553 `ERROR: Error writing to output file: [filename].`

Explanation

An error has been encountered in writing to the output file. This could be caused by a variety of environmental factors (such as a lack of disk space). **`stderr`** may contain library messages with additional information about the error.

554 `ERROR: Inline member function does not end.`

Explanation

The end of the input file has been encountered before the closing brace (}) was seen for a member function.

555 `ERROR: Static function '[function-name]' was not defined.`

Explanation

The specified function was declared **`static`** but has not been defined in this source file.

556 `ERROR: Global anonymous unions must be static.`

Explanation

An anonymous union declared at file scope must be declared as **`static`**.

557 `ERROR: Anonymous unions may not have function members.`

Explanation

You cannot define an anonymous union that contains function members. If you want function members, use another construct, such as a plain **`union`**, **`struct`**, or **`class`**.

558 `ERROR: Anonymous unions may not have private or protected members.`

Explanation

You cannot define an anonymous union that contains private or protected members. If you want private or protected members, use another construct, such as a plain **`union`**, **`struct`**, or **`class`**.

559 **ERROR: '[identifier]' redeclared in anonymous union.**

Explanation

The specified identifier was previously declared to have some other type.

560 **ERROR: An anonymous union cannot be declared as a static member.**

Explanation

An anonymous union has no name for linkage to the definition.

561 **ERROR: Cannot load special character table '[table-name]'.**

Explanation

The character set translation table could not be found, was corrupted, or was installed incorrectly.

562 **ERROR: Conflicting declaration of name '[identifier]' reserved for [purpose].**

Explanation

You have inadvertently used a name that the translator or compiler reserves for its own uses. Choose another name.

564 **ERROR: Cannot initialize a function '[function-name]'.**

Explanation

You cannot initialize a function. You can initialize only variables.

565 **ERROR: Static members [member-name] cannot be initialized by a mem-initializer.**

Explanation

Static members should be initialized by the definition of the **static** member outside the class. For example, this message is issued if you use the following code and a is a **static** member:

```
X::X() : a(10)

{ . . . }
```

566 **ERROR: Enum constants [identifier] cannot be initialized by a mem-initializer.**

Explanation

Enum constants should be initialized inside the **enum** declaration, as they are in C. For example, this message is issued if **a** is an enumeration constant:

```
X::X() : a(10)

{ . . . }
```

567 `ERROR: Types must match in a destructor expression: (<object-type>).~<destructor-type>`

Explanation

When calling a dummy destructor for a builtin type, it is required that the type specified in the destructor name match the type (dereferenced for `->`) of the left operand of the `.` or `->` expression. This message is also output for class destructors when the destructor is specified as an unqualified name (i.e. `''object.~type_name''`) and the specified type does not match the object type.

568 `ERROR: Cannot create a new value of a void.`

Explanation

The `new` operator cannot be applied to `void`. Because `void` is not an object type, a pointer to it could not be returned by `operator new`.

569 `ERROR: Loop in -> operators.`

Explanation

The pointed-to expression uses a user-defined operator -> that either returns the class that contains the operator or returns a class that contains another operator -> which in turn returns the class containing the original operator ->. The loop might be more complicated, but in any event the sequence leads back to the original operator ->.

570 `ERROR: A linkage-specification may occur only in file scope.`

Explanation

Linkage-specifications are not permitted in block scopes, class scopes, or function scopes.

571 `ERROR: Cannot define a type in return or argument types.`

Explanation

A type (for example a `struct` tag) cannot be defined in an argument list or in the specification of the return type.

572 `ERROR: [object] may not have the same name as its class.`

Explanation

A `static` data member, enumeration, member of an anonymous union, or a nested type cannot have the same name as its class.

573 `ERROR: An overloaded operator '[operator]' cannot have default arguments.`

Explanation

It is illegal to declare overloaded operators with default arguments. For example, `int operator + (int=3,int=4)` is not a valid declaration.

574 `ERROR: Invalid use of abstract class: [class-name].`

Explanation

An abstract class cannot be used as an argument type, a function return type, or the type of an explicit conversion.

575 `ERROR: An object of a class with an [object-type] may not be a member of a union.`

Explanation

An object of a class with constructors, destructors, or user-defined assignment operators cannot be a member of a `union`. For more information, refer to CPL2 r.9.5.

576 `ERROR: Error declaring 'new': [reason].`

Explanation

`operator new` must have a return type of `void*`. Its first argument is required and must be of type `size_t`. For more information, refer to CPL2 r.12.5.

577 `ERROR: Error declaring 'delete': [reason].`

Explanation

The `delete` function must have return type `void`. Its first argument must be of type `void*` and if there is a second argument, it must be of type `size_t`. No more than two arguments are permitted. For more information, refer to CPL2 r.12.5.

578 `ERROR: Initializer-clause cannot be used for class having an [object-type].`

Explanation

A class having a constructor, a private or protected member, a base class, or a virtual function is not an aggregate and cannot be initialized by means of an initializer-clause (for example, `={10,2,10.2}`). For more information, refer to CPL2 r.8.4.1.

579 `ERROR: Conversion to a virtual base class '[class-name]' from a derived class is not allowed for member pointers.`

Explanation

Virtual base classes cannot be converted, explicitly or implicitly, from derived classes.

580 `ERROR: Cannot return [attempted-return-type] from function returning [declared-return-type].`

Explanation

The return value is of a type that cannot be converted to a type required by the function's return type.

581 `ERROR: Function '[function-name]' has an initializer.`

Explanation

Functions cannot be initialized, although function pointers can be initialized.

582 `ERROR: Character array [array-name] too short for string of length [string-length].`

Explanation

A character array cannot be initialized by a string that has more characters than the array has elements.

583 `ERROR: Too many initializers for [array-name]: found [num] initializers.`

Explanation

No array can be initialized with more initializers than the array has elements.

584 `ERROR: Too many initializers for [class-name].`

Explanation

No class can be initialized with more initializers than the array has members.

585 `ERROR: Left operand of '[operator]' must be [type].`

Explanation

The left operand of the dot operator (.) must be a class object.

586 **`ERROR: Type '[type]' is invalid for the left operand of '[operator]'.`**

Explanation

The left operand of the arrow operator (->) must resolve to a class pointer.

587 **`ERROR: Case label value must be a constant expression.`**

Explanation

You have used a non-constant expression as a **`case`** label.

588 **`ERROR: Duplicate case label value.`**

Explanation

The same **`case`** label occurs more than once within a **`switch`** statement.

589 **`ERROR: More than one default.`**

Explanation

There is more than one **`default`** label in a single **`switch`** statement.

590 **`ERROR: [symbol-name] is not an enum.`**

Explanation

The specified symbol was used after the **`enum`** keyword, but is not an enumeration.

591 **`ERROR: [symbol-name] is not a class, struct, or union.`**

Explanation

The specified symbol was used after a **`class`**, **`struct`**, or **`union`** keyword, but is not a **`class`**, **`struct`**, or **`union`**.

592 **`WARNING: Wide and narrow character strings concatenated, using [width].`**

Explanation

Wide characters strings are of the form **`L''abc''`**; narrow characters strings are the usual **`''abc''`**. These two types of strings should not be concatenated together. For example, neither of the following statements are valid:

```
"abc" L"def" /* wrong */

L"abc" "def" /* wrong */
```

If the first string in the concatenation is wide, the translator treats the result as a wide string. Similarly, if the first string is narrow, the translator treats the result as narrow.

593 `WARNING: Missing return statement.`

Explanation

A **`return`** statement is missing at the end of the outer block of a function and a return value is required.

594 `WARNING: Zero-length array used.`

Explanation

Zero-length arrays are allowed in classes (types **`class`**, **`struct`**, and **`union`**) as an extension of standard C and C++. This message is only to warn you about the use of this extension. If you do not want to see this warning, use the **`suppress`** translator option to turn it off.

595 `ERROR: Floating point value out of range.`

Explanation

The value of a floating-point constant, or the result of an operation on floating point constants is either too large or too small to be represented. Frequently, this is caused by a multiplication or division of two floating-point constants.

596 `ERROR/WARNING: [data-type-1] initialized with [data-type-2].`

Explanation

A shorter data type has been initialized with a longer data type. For example, an **`int`** has been used to initialize a **`char`**. If the initializer is a constant whose value is too large to be represented by the shorter type, this is an error. If the initializer is a variable, this is a warning. (If the value of a constant initialization can be represented by the shorter type, neither a warning nor an error is issued.)

597 `ERROR/WARNING: [data-type-1] assigned to [data-type-2].`

Explanation

A longer data type has been assigned to a shorter data type. For example, a **`long`** has been assigned to a **`short`**. If the value assigned is a constant which is too large to be represented by the shorter type, this is an error. If the assigned type is a variable, this is a warning. (If the value of an assigned constant can be represented by the shorter type, neither a warning nor an error is issued.)

598 **ERROR/WARNING: [sign-type-1 data-type-1] assigned to [sign-type-2 data-type-2].**

Explanation

A signed data type has been assigned to an unsigned data type of the same size (for example, a **long** has been assigned to an **unsigned long**), or conversely, an unsigned data type has been assigned to a signed data type of the same size (for example, an **unsigned int** has been assigned to an **int**). If the assigned value is a constant that cannot be represented correctly by the type to which it is assigned, an error results. Otherwise, the message is a warning.

599 **ERROR: A function definition [name] requires an explicit parameter list.**

Explanation

The named function was defined with a function typedef name. This is erroneous because the parameter list is not explicitly stated in the function definition. For example, the following code would produce this message:

```
typedef int F(int);
F foo
    {
    return 10;
    }
```

600 **WARNING: Implicit conversion of member function [name] to a member pointer.**

Explanation

This warning indicates that a non-standard feature was specified. The address-of (&) operator is implicitly applied by the compiler to the named member function to create a member pointer to function. Other C++ systems may require the address-of operator.

601 **WARNING: Symbol [name] was not used.**

Explanation

This warning is given for symbols local to a function that are never used and sometimes indicates a mistake in the function.

602 **`WARNING: Variable [name] used before being initialized.`**

Explanation

An uninitialized variable was used before it was assigned.

603 **`WARNING: Symbol [symbol] was initialized but never referenced.`**

Explanation

This warning is given for symbols local to a function that are defined but that do not have their value used. This warning sometimes indicates a mistake in the function.

604 **`ERROR: Return type of 'operator [new | delete]' cannot be changed after it has been used.`**

Explanation

C++ pre-declares library versions of the default operator **`new`** and operator **`delete`** functions. A program may declare its own versions of these operator functions with differing return types from those in the library, but this must be done before any use of these operator functions.

Action

This error can be fixed by moving the declarations of **`new`** and **`delete`** before using their operators.

605 **`ERROR: [name] must be a function, not a [type].`**

Explanation

An overloaded operator must be a function, not a type.

606 **`ERROR: [data-type] may not use ellipsis ('...').`**

Explanation

The ellipsis punctuator ('...') may not be used in an overloaded operator except with operator **`new`**.

607 **`ERROR: [name] must return a [type1] not a [type2].`**

Explanation

The named function must return a [type1] not a [type2].

608 `ERROR:'operator new's first argument must be a 'size_t'.`

Explanation

Operator **`new`** must be a function returning a void pointer. Its first argument must be a **`size_t`**.

609 `ERROR: 'operator delete's first and only argument must be a 'void *'.`

Explanation

Operator **`delete`** must be a function returning void. Its first and only argument must be a void pointer.

610 `ERROR: Previous declaration of [name] was [attribute1], this declaration is [attribute2].`

Explanation

An attribute of a symbol was declared differently in a previous declaration. The message specifies the symbol and attribute.

Note: For some attributes, C++ will apply a default if the attribute is not explicitly specified in the declaration. A declaration with a specific keyword may conflict with a previous declaration with no keyword, depending on the current defaults. Which defaults are applied by C++ depend upon the specific attribute and the user supplied options. For example, if a previous declaration specified **`__rent`**, the current declaration for the same symbol cannot specify **`__norent`**. Declarations must be consistent.

611 `ERROR: Previous declaration of [name] differed in the use of [attribute].`

Explanation

An attribute of a symbol was declared differently in a previous declaration. Either the previous declaration used a keyword that is not present in this declaration, or did not use a keyword that is present in this declaration.

Note that for some attributes, C++ will apply a default if the attribute was not explicitly specified in the declaration. So a declaration with a specific keyword may conflict with a previous declaration with no keyword depending on the current defaults. Which defaults are applied by C++ depend upon the specific attribute and the user supplied options. The declarations should be made consistent.

617 `ERROR: Default argument(s) of function used before they are defined.`

Explanation

The default arguments of one function cannot use the default arguments of a function declared later in the class.

618 `ERROR: Invalid use of K&R C style declaration for [name].`

Explanation

Only global functions may use the K&R C style function syntax. Member functions must use the C++ prototype syntax. This error sometimes occurs because of a syntax error in the formal argument list.

619 `ERROR: Bitfields cannot be static members.`

Explanation

Bitfields cannot be static members.

Action

Either make the bitfield non-static, change the bitfields to an integral type, or nest the bitfield in another **`struct`**, and make the struct a static member.

Reference

Stroustrup, Bjarne (1991), *The C++ Programming Language*, Reading, MA: Addison-Wesley Publishing Company.

620 `ERROR: Missing template argument list for '[template-name]'.`

Explanation

The specified template name was used without a template argument list in a context where the argument list is required. For class templates, the argument list is required except within declarations of the template class itself.

621 `ERROR: Invalid template argument.`

Explanation

A template argument must be a constant expression (not floating point), the address of an object or function with external linkage, or the name of an object or function with external linkage.

622 `ERROR: Not enough arguments for template.`

Explanation

The template argument list had too few arguments specified.

623 `ERROR: Too many arguments for template.`

Explanation

The template argument list had too many arguments specified.

624 `ERROR: Cannot instantiate incomplete template class '[template-class]'.`

Explanation

A use of a template class requires its definition, but the definition of the template has not been completed.

625 `ERROR: Recursive template instantiation of '[template-class]'.`

Explanation

During instantiation of a template class specialization, a use of the class was encountered that required it to already be instantiated.

626 `ERROR: Template actual parameters cannot depend on unnamed or local types.`

Explanation

A template argument contains a user defined type that is either unnamed, declared inside a function, or declared inside an unnamed scope.

627 `ERROR: Template instantiation failed for '[template]'.`

Explanation

Instantiation of the template specialization could not be completed. This could be due to a syntax error diagnosed in preceding messages.

628 `WARNING: Template function not deducible.`

Explanation

Not all template argument types and values can be determined from the function parameter types. References to this template function must explicitly specify the non-deducible template arguments.

630 **ERROR: Template declaration has a nested class definition.**

Explanation

A template declaration defines a class, but it also includes a declarator. The class definition should be declared separately from the item referred to by the template declarator.

```
template <class T>
   class C { . . . }* func( T* ); // Error
```

631 **ERROR: Scope for '[name]' does not match a class template.**

Explanation

The scope for the name being declared in a template must match a class template. For example:

```
template <class T, class U> class C {
   class Nested { ... };
   ...
};

template <class X, class Y> int C<X, Y>::i = 1; // OK
template <class X, class Y> int C<X, Y>::Nested::i = 1; // OK
template <class X, class Y> int C<Y, X>::j = 0; // Error
template <class X, class Y> int C<Y, int>::k = 7; // Error
```

In other words, the template parameters in effect must match the parameters from the template class declaration and the argument list for the template scope must use the template parameters in the original order.

632 **ERROR: Template class member declaration must define the member.**

Explanation

Template declarations of template class member functions must define the member, except for friend declarations. The same restriction exists for non-template class members.

633 **WARNING: Statement has no effect.**

Explanation

The evaluation of the specified statement has no side-effects.

634 `ERROR: Unrecognized template declaration.`

Explanation

A template declaration must declare or define a function, class, or static data member.

635 `ERROR: Template argument list does not match formals.`

Explanation

A template argument does not match the corresponding template formal. This could occur if, say, a type argument was specified for a data formal, or a data argument was specified for a type formal.

636 `ERROR: Multiple definitions for '[symbol]'.`

Explanation

This indicates that specializations of multiple template member definitions match the given member function or data item.

637 `ERROR: Template actual '[object]' does not have external linkage.`

Explanation

The object or function denoted by a template argument expression must be declared so that it can be referred to by name in other translation units.

638 `ERROR: Ambiguous template declarations for '[function-signature]'.`

Explanation

This indicates that two or more distinct template declarations correspond to the given function signature. It may be possible to disambiguate the template by explicitly specifying template arguments.

639 `ERROR: Type members ([name]) cannot be initialized by a mem-initializer.`

Explanation

A mem-initializer specified the name of a type member (i.e. a **`class`**, **`enum`**, or **`typedef`** name) whose type is not a base class of the class being constructed.

640 `ERROR: Invalid scoped declaration in member list.`

Explanation

The only member declarations that allow scoped name declarators are access and `friend` declarations.

641 `ERROR: Illegal bitfield declaration.`

Explanation

Bitfields may only be declared for non-static class members. In particular, `typedef` may not be used with a bitfield declaration.

642 `ERROR: Templates may not be declared in local scopes.`

Explanation

Templates must be declared outside any function.

643 `ERROR: Invalid explicit template specialization.`

Explanation

An explicit template specialization declaration (which begins with `template <>`) must refer to a specialization of a previously declared template function, template class, or static member of a template class.

644 `ERROR: Template function was explicitly specialized after its first use.`

Explanation

An explicit template specialization refers to a function that has been previously used. The explicit specialization should be moved before the first use.

645 `WARNING: Old-style template specialization.`

Explanation

The draft C++ standard requires explicit specializations of template functions, classes, and members to be introduced with `template <>`.

646 **ERROR: Template class was specialized after it was implicitly instantiated.**

Explanation

An explicit specialization of an instance of a template class was provided after a use of the specialization that required it to be implicitly generated from its template. The explicit specialization should be moved before the first use of the specialized class that requires the class to be implicitly instantiated.

647 **ERROR: Invalid explicit template instantiation.**

Explanation

An explicit template instantiation declaration (which begins with the keyword **template** but has no following template formal list) must specify a specialization of a previously declared template function, template class, or static member of a template class.

648 **ERROR: No template definition for '[template]'.**

Explanation

An explicit template instantiation was requested for a template specialization (of a template function, template class, or template class member) which does not have a template definition.

649 **ERROR: Template instantiations are too deeply nested (>[number] levels).**

Explanation

Instantiation of template items required too many pending levels of instantiation. The instantiation process may be unbounded. For example:

```
template <class X> class C { C<X*> c; };

C<int> x;  // requires C<int>, C<int*>, C<int**>, C<int***>, ...
```

650 **ERROR: Template item was both explicitly instantiated and specialized.**

Explanation

Explicit instantiation was requested for a template specialization which was already specialized. Either the member was explicitly specialized or the containing class was specialized.

651 **`ERROR: Explicit instantiation of a compiler generated function.`**

Explanation

An explicit instantiation may not refer to an compiler created member of a template class (i.e. implicitly declared constuctors, destructors, and assignment operators).

652 **`ERROR: Static data member of a template was specialized after it was used.`**

Explanation

The explicit specialization must precede the first use of the static member that requires a definition.

653 **`WARNING: Keyword '[keyword]' is not legal in function declaration.`**

Explanation

The keyword was not legal in the declaration. For example, **`extern`**.

654 **`ERROR: Invalid scoped declaration.`**

Explanation

A scoped declaration was found in a context where it was not expected. A declaration for a scoped name must refer to a previously declared member of the specified scope.

655 **`ERROR: Unexpected use of template function name.`**

Explanation

A template function name (like "f<int>") was used in a context where it was unexpected, such as a non-function declaration. Explicit specializations of functions explicitly specifying a template argument list should begin with **`template <>`**.

656 **`ERROR: Unexpected initializer in an explicit template instantiation declaration.`**

Explanation

Explicit template instantiations should not specify an initializer.

657 **`ERROR: Elaborated name depends on a template parameter.`**

Explanation

Elaborated names which contain template parameters are not allowed in template function declarations. For example:

```
template <class T>
void f( class T* );  // Error
```

658 **ERROR: A template declaration may declare only one item.**

Explanation

Multiple template functions or data members may not be declared in the same template declaration. For example:

```
template <class T>
void f( T* ),
     g( T* );  // Error
```

---

659 **ERROR: Unnamed class definition in a template declaration.**

Explanation

Classes defined in a template declaration must be named. For example:

```
template <class T>
class { . . . }; // Error
```

---

660 **ERROR: Template formal parameters cannot [type].**

Explanation

Template formal parameters cannot be floating point, be function member pointers, be void, have function type, or have class type. A template data formal parameter has an invalid type.

---

661 **ERROR: Instantiation of '[name]' has type ([type]) but [type] was expected.**

Explanation

Instantiation of **[name]** has **[type]**, but a class type was expected.

A template declaration, when specialized, used a template formal dependent name in a context where a class name was expected, but the specialized type was not a class. For example:

```
template <class T> void f( int T::* p );

void testit()
{
   f<int>( 0 );  // Error
}
```

---

662 **ERROR: Instantiated name '[name]' does not resolve to [type/scope].**

Explanation

Instantiated **[name]** does not resolve to a **[type/scope]** name.

A template declaration, when specialized, used a template formal dependent name in a context where a scope or type name was expected, but the specialized type was not a class type. For example:

```
template <class T>
class C { public: int Z; };

template <class U>
void f( typename C<U>::Z* p );

void testit()
{
   f<int>( 0 ); // Error, C<int>::Z is not a type
}
```

663 **ERROR: Incomplete template member initializer. Found unexpected '[token]'.**

Explanation

While scanning the syntax for an initializer in a template member declaration, an unexpected token was found. This probably indicates a missing closing parenthesis or brace. To improve error detection, syntactically valid but semantically invalid constructs like embedded class definitions are not allowed inside the initializer.

For example:

```
template <class T> struct C {
   static int si;
};
template <class T>
int C<T>::si = ( 5 ;  // unexpected ';' due to missing ')'
```

The C++ translator can sometimes be confused by commas in nested template argument lists in the initializer. Enclosing the initializer in parentheses avoids this limitation:

```
template <class T>
int C<T>::si = ( otherTemplate< T, T>().memFunc() );
```

664 **ERROR: Template function specialization does not correspond to any template declarations.**

Explanation

The declaration of a template function specialization does not match any of the function template declarations.

665 `WARNING: Missing elaborator on a friend class declaration.`

Explanation

The declaration of a class as a friend should use an elaborated name.

667 `ERROR: Template member declaration does not match the member '[member-name]'.`

Explanation

The template declaration of a template class member, when instantiated, did not match the declaration from the template class.

668 `ERROR: Default arguments are not allowed on template redeclaration '[template]'.`

Explanation

Function templates must declare all their template default arguments on their first declaration. Out-of-line template definitions of template class member functions may not declare default arguments. Explicit specialization and instantiation declarations may not declare default arguments.

669 `WARNING: Dubious type specifier '[qualified-name]'. Use 'typename'`

Explanation

While parsing a template declaration, a qualified name whose scope depends on template formals was encountered where a type specifier was expected. The name is assumed to represent a type, however the draft C++ standard requires a **`typename`** keyword to prefix the qualified name. This warning could also occur because of missing type specifiers in a declaration.

670 `WARNING: Missing return type specifier.`

Explanation

The draft C++ standard requires that function declarations (except for constructors, destructors, and type conversion operators) specify a return type. The "implicit **`int`**" rule of C is considered obsolescent.

672 `ERROR: A condition cannot specify a function or an array.`

Explanation

The declarator in a condition that is an initialized declaration cannot declare a function or an array.

673 `ERROR: A condition cannot declare a new class or enumeration.`

Explanation

The declarator in a condition that is an initialized declaration cannot define a new class or enumeration.

674 `ERROR: No such [conversion-specification] conversion; to ( [type] ) from ( [type] ).`

Explanation

The requested conversion was not one of the conversions allowed by the specified new-style cast operator. For example, you would get this error if you used `static_cast` to try to convert a pointer to an integral type.

675 `ERROR: Invalid qualifier change; to ( [qualifier] ) from ( [qualifier] ).`

Explanation

The requested new-style cast conversion either discarded qualifiers or was not "const-safe". If the qualifier change is desired, use `const_cast<>` to perform the needed qualifier conversion.

676 `ERROR: Cannot apply typeid operator to an undefined class.`

Explanation

If the expression or type name operand has class or reference to class type, the class definition must be complete.

677 `ERROR: Cannot apply typeid operator to an overloaded function name.`

Explanation

The function name must be resolved to a specific function before using it.

678 `WARNING: Polymorphic type information may not be available.`

Explanation

`typeid` or `dynamic_cast` was applied to a polymorphic type, but the `rtti` translator option was not specified.

679 `ERROR: Header <typeinfo> must be included before using typeid.`

Explanation

The `<typeinfo>` header must be included at global scope before the first use of the `typeid` operator in order to declare the `type_info` class.

680 `ERROR: Dynamic_cast from undefined class pointer not allowed.`

Explanation

The class referred to by the source pointer value must be previously defined.

681 `ERROR: Polymorphic class object required.`

Explanation

When using **`dynamic_cast`** to convert to a void pointer type, the source type must be a pointer to a polymorphic class type. (A polymorphic class is one that has virtual function members.) When using **`dynamic_cast`** to convert to a class pointer type, the source type must be a pointer to the same class, a pointer to a derived class of the target class, or a pointer to a polymorphic class. When using **`dynamic_cast`** to convert to a class reference, the source type must be of the same class, a derived class of the target class, or a polymorphic class.

682 `ERROR: Invalid use of type member [member].`

Explanation

The name specified as the right operand of a `.` or `->` operator must be a data or function member.

683 `WARNING: Implicit template instantiation uses static object [object].`

Explanation

During implicit template instantiation, a use of a static data or function item was detected. Such uses refer to different objects in each compilation unit, violating the requirements for automatic template instantiation. The static use should be removed, or the template item should be explicitly specialized.

684 `WARNING: Static use forces automatic template instantiation in primary module.`

Explanation

This message is output when static object use (diagnosed by message 683) forces an automatically instantiated item to be inserted in the primary output module for the current compilation unit. This is because static items are only available in the single output module that defines them.

685 **`WARNING: Inline function [function] uses static object [object].`**

Explanation

This message is output when the definition of a non-static inline function (either with external linkage or contained in an inline function with external linkage) uses a static object. The function cannot be defined in other compilation units without breaking the one definition rule of C++, and it cannot be used in an automatically instantiated template without producing message 684. The function will be treated as a static object for the purposes of messages 683 to 685 if automatic template instantiation is turned on.

686 **`ERROR: Inline function [function] was used but not defined.`**

Explanation

The specified function was declared inline and used, but it was never defined in this source file.

687 **`WARNING: Dangling temporary reference.`**

Explanation

This message is output when the compiler detects that a reference has been bound to a temporary whose lifetime is shorter than the lifetime of the reference. Such code is likely to be erroneous. Cases in which this warning is produced include the binding of a temporary to a local static reference:

```
void example()
{
   static const int& i = 5;
 // temporary holding the value 5 lives until function
 // return but the reference persists and is not rebound
 // the next time example() is called.
}
```

688 **`ERROR: A condition initializer must be an expression.`**

Explanation

The initializer for a condition declaration must be an expression, not a brace-enclosed list initializer.

689 `Ambiguous result for conditional expression.`

Explanation

The second and third operands of a conditional expression are not the same type. At least one is a class value. Each of the operands can be converted to match the type of the other operand.

To avoid this problem, use `static_cast` to be explicit about the intended conversion.

---

690 `Invalid use of non-static member function.`

Explanation

An expression that designates a non-static member function can be used in limited contexts. A non-static member function name can be used as one of the following:

- a member pointer constant
- the name following . or ->
- the left operand of a call expression.

Other expressions that designate non-static member functions (such as the result of expressions with the following operators:)

.

->

.*

->*

can be used only as the left operand of a call expression.

This error could indicate a missing call argument list or an incorrectly specified member pointer constant.

---

691 `Ambiguous use of overloaded function name '<function-name>'.`

Explanation

The specific function could not be resolved because an overloaded function name was used in a context other than the function expression of a function call. Besides function expressions, overloaded function names are resolved (by using the target type) in only the following contexts:

- As a parameter to a function or user operator.
- As an initializer value.
- As the right-hand size of an assignment.
- As a return value.
- As the operand of an explicit cast or `static_cast`.

The ambiguity can be resolved by using an explicit cast to select a particular function.

---

9 LSCX Run-time Messages

Library Message Processing

The SAS/C run-time library generates messages for unusual conditions detected during program execution. Some of these conditions represent programmer errors, such as attempting to take the square root of a negative number, while others represent unusual conditions beyond the programmer's control, such as running out of memory or a system service failure.

The library does not diagnose every "failure" of every function because some failures are expected. Rather, it is the application's responsibility to detect these conditions and inform the user when necessary. For example, the library will not diagnose a failure to retrieve an environment variable that is not defined because the caller of **getenv** should be prepared for this possibility and usually will not consider it to be an error.

If the library produces diagnostics that are not desirable for your application, you can use

- the **quiet** function to suppress these messages, as described in *SAS/C Library Reference, Volume 1*
- the **=warning** run-time option to force library diagnostics to be displayed, even if suppressed by **quiet**
- the **=btrace** run-time option to get a list of the active functions (a *traceback*) at the time a diagnostic is generated.

Message Types

Run-time library diagnostic messages have the form

```
LSCX[num] **** [severity] **** ERRNO = [errno value]
        Generated in [function] called from line [num] of [function],
                     offset [hex]
        [C++/Extended] name: [fullname]
        [message text]
        Interrupted while: [context]
```

where [severity] can be any of the following:

NOTE	describes a condition that permits program execution to continue but which is not communicated to the caller of the routine; **errno** is usually not set.
WARNING	describes a condition that permits program execution to continue; however, the routine that detected the condition returns an error indication to its caller. When a library WARNING is issued, the **errno** variable is set and usually an error code is returned from the function that detected the condition. (Most library messages are WARNINGs.)
ERROR	describes a condition that forces program termination, usually with an ABEND.

Run-time library messages are normally written to the **stderr** file, except for ABEND messages, which are written to the terminal, the JES2 job log or to an MVS SYSOUT file. (See message 064 for more information about ABEND messages in batch.) If **stderr** is not open or not usable, library messages will be sent to the terminal or to the JES2 job log in MVS batch.

Note: Messages numbered from 000-099 are issued under special circumstances and are not usually in the format described above; they never modify **errno**. Some of these messages are also system dependent. Messages numbered between 500 and 999 often refer to system-specific conditions, and the same number may be used by different systems in different ways. When this occurs, the same message number will be listed several times, once for each system where it has a unique meaning.

For additional assistance with any diagnostic, call Technical Support at SAS Institute. In preparation for your call, read Technical Report C-114, *A Guide for the SAS/C Compiler Consultant* and have ready the required information.

LSCX Messages

000 **ERROR: Automatic storage overlaid by program.**
ABEND code: 1200

Explanation

During return from a function, the library detected that stack storage for the returning function was inconsistent or corrupted. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length.

Action

Run the program again using the **=storage** run-time option, or run the program under the SAS/C debugger and use the **storage** debugger command to locate the overlaid data areas. Also, inspect the returning function for uninitialized pointers, out-of-bounds array references, and other similar errors.

---

001 **ERROR: Automatic storage control blocks overlaid.**
ABEND code: 1201

Explanation

During a function call, the library detected that stack data areas were inconsistent or corrupted. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length.

Action

Run the program again using the **=storage** run-time option, or run the program under the SAS/C debugger and use the **storage** debugger command to locate the overlaid data areas. Also, inspect the calling function for uninitialized pointers, out-of-bounds array references and other similar errors.

---

002 **`ERROR: Stack overflow detected.`**
`ABEND code: 1202`

Explanation

During program termination, the library discovered that the end of the stack had been overlaid. This error can occur only when a program runs with the **`=minimal`** run-time option. Note that stack overflow is not detected when it occurs but only at program termination. However, if stack overflow is detected, then all the program's results should be regarded as possibly incorrect.

Action

Run the program again without specifying **`=minimal`**, and use the **`=usage`** run-time option to obtain an accurate estimate of the amount of stack space required by the program. Then adjust the initial stack space requested accordingly, adding some extra space for error handling and other unusual situations.

---

003 **`ERROR: Automatic storage control blocks overlaid.`**
`ABEND code: 1203`

Explanation

During program termination, the library discovered that stack management data areas were inconsistent or corrupted. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length. Note that even though this error is detected during termination, it may have occurred much earlier, and all the program's results should be regarded as possibly incorrect.

Action

Run the program again using the **`=storage`** run-time option, or run the program under the SAS/C debugger and use the **`storage`** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

004 **`ERROR: Improper or outdated argument to longjmp.`**
`ABEND code: 1204`

Explanation

During a call to **`longjmp`**, the library was unable to locate the function to which control was to be returned. This may be caused by overlaying a **`jmp_buf`**. It may also be caused by calling **`longjmp`** for a **`jmp_buf`** set by a call to **`setjmp`** in a routine that has already returned to its caller.

Action

Correct the problem. If a **`jmp_buf`** has been overlaid, the debugger **`monitor`** command may be useful in locating the cause of the overlay.

---

005 `ERROR: Free storage control blocks overlaid.`
`ABEND code: 1205`

Explanation

During execution of the **`malloc`** function, the library detected that library free storage control blocks had been overlaid. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length. Note that the overlay may have occurred much earlier in the program's execution, and may have nothing to do with the code executing at the time of the ABEND.

Action

Run the program again using the **`=storage`** run-time option, or run the program under the SAS/C debugger and use the **`storage`** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

006 `ERROR: Free storage control blocks overlaid.`
`ABEND code: 1206`

Explanation

During execution of the **`malloc`** or **`free`** function, the library detected library free storage control blocks had been overlaid. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length. Note that the overlay may have occurred much earlier in the program's execution, and may have nothing to do with the code executing at the time of the ABEND.

Action

Run the program again using the **`=storage`** run-time option, or run the program under the SAS/C debugger and use the **`storage`** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

007 `ERROR: Free storage control blocks overlaid.`
`ABEND code: 1207`

Explanation

During execution of the **`free`** function, the library detected that library free storage control blocks had been overlaid. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length. Note that the overlay may have occurred much earlier in the program's execution, and may have nothing to do with the code executing at the

Action

Run the program again using the **`=storage`** run-time option, or run the program under the SAS/C debugger and use the **`storage`** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

008 **ERROR: Storage to be freed is unallocated or overlaid.**
ABEND code: 1208

Explanation

The argument of a call to **free** did not appear to be the address of storage previously allocated by **malloc**. One possible explanation is that the 8-byte prefix that precedes each block of storage allocated by **malloc** has been overlaid. Note that the overlay may have occurred much earlier in the program's execution and may have nothing to do with the code executing at the time of the ABEND.

Action

If the storage to be freed appears to have been correctly allocated with **malloc**, run the program again using the **=storage** run-time option, or run the program under the SAS/C debugger and use the **storage** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

009 **ERROR: Free storage control blocks overlaid.**
ABEND code: 1209

Explanation

During program termination, it was discovered that free storage management data areas were inconsistent or corrupted. This is probably caused by the program storing data using an invalid pointer, a bad subscript, or an incorrect string length. Note that even though this error is detected during termination, it may have occurred much earlier, and all the program's results should be regarded as possibly incorrect.

Action

Run the program again using the **=storage** run-time option, or run the program under the SAS/C debugger and use the **storage** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

010 **ERROR: Program terminated by call to abort.**
ABEND code: 1210

Explanation

The program has called the **abort** function or raised the **SIGABRT** signal. This may occur as the result of a failed assertion when the **assert** macro is used.

Action

Determine the reason that **abort** was called, and correct the problem. Note that if this message occurs running the compiler or a SAS/C utility, it indicates a programming error in the compiler or utility and should be reported to SAS/C Technical Support.

---

011 `ERROR: Argument to unloadm is not a loaded function address.`
`ABEND code: 1211`

Explanation

The program called the `unloadm` function, but the argument did not appear to be a pointer to a function in a load module loaded by `loadm`. One possible cause of this message is an overlay of the function pointer.

Action

If the argument to `unloadm` appears to be a valid function pointer into a module loaded by `loadm`, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors. Use of the debugger `monitor` command may also be helpful in locating the overlay.

---

012 `ERROR: Unable to load run-time I/O routines, execution cannot continue.`

012 `ERROR: Unable to locate transient library.`

012 `ERROR: Unable to locate diagnostic routines, execution cannot continue.`
`ABEND code: 1212`

Explanation

These messages indicate that required transient library modules could not be loaded. The first message above is the most common case. This message occurs during program initialization if the library's I/O and error-handling routines could not be found. This most likely indicates a problem with JCL, TSO allocations or CMS disk accesses, such that the program is unable to access the transient library at all. The second message is produced when certain transient modules (for instance, signal handling) cannot be located after initialization is complete. This form of the message most likely indicates an error installing the transient library. The third form of the message occurs only when the Generalized Operating System interface (GOS) is in use.

Action

Confirm that the transient library is available. Contact your local SAS/C support to determine the proper procedures for access to the transient library at your site.

In MVS batch or TSO, the library startup code looks for the transient library modules in one of the following places:

1. In the library allocated to the DDname CTRANS.
2. In any allocated task library. (For instance, under ISPF, the DDname ISPLLIB will be searched.)
3. In STEPLIB or JOBLIB.
4. In the system linklist.
5. In the MVS link pack area (LPA).

If the library has not been installed into linklist or LPA, any SAS/C job or session must have an appropriate DD statement defined as in items 1 through 3 above.

In CMS, the library startup code looks for the transient library modules in one of the following places:

1. In a shared segment created when the product was installed.
2. In LSCRTL LOADLIB on an ACCESSed minidisk.

If the library has not been installed into a segment or segments, you must access the library minidisk before running a SAS/C program.

Under OpenEdition MVS, when a program is invoked by the **exec** system call, the library startup code looks for the transient library modules in one of the following places:

1. In the library named by the **ddn_CTRANS** environment variable.
2. In the list of libraries defined by the **STEPLIB** environment variable.
3. In the system linklist.
4. In the MVS link pack area (LPA).

If the library has not been installed into linklist or LPA, any SAS/C program run under OpenEdition will require definition of one of the environment variables listed above.

For a C program linked using a Generlized Operating System library, where the library looks for transient modules is not defined by SAS/C. In this case, consult the implementor of the GOS interface routines.

---

013 **ERROR: Improper use of C subordinate load module.**
ABEND code: 1213

Explanation

An attempt was made to run a SAS/C dynamically loadable module as an independent program. A SAS/C dynamically loadable module can only be used when loaded by another SAS/C program using the **loadm** function.

Action

Refer to the *SAS/C Library Reference*, Chapter 1, "Dynamic-Loading Functions."

---

014 **ERROR: Recursive failures in run-time diagnostic routines.**
ABEND code: 1214

Explanation

The SAS/C library diagnostic support suffered multiple errors while trying to write a diagnostic message. This is probably a SAS/C library error, though it could be caused by program overlays of library data areas.

Action

Contact SAS/C Technical Support.

---

015 `ERROR: Load module management control blocks overlaid.`
`ABEND code: 1215`

Explanation

The CMS `unloadm` function detected an error trying to delete the storage used by a dynamically loaded load module. This is probably caused by program overlay of library data areas.

Action

Inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

016 `ERROR: Load module management control blocks overlaid.`
`ABEND code: 1216`

Explanation

The CMS `unloadm` function discovered that library dynamic loading control blocks were inconsistent or corrupted. This is probably caused by program overlay of library data areas.

Action

Inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

017 `ERROR: Load module management control blocks overlaid.`
`ABEND code: 1217`

Explanation

The CMS `loadm` function discovered that library dynamic loading control blocks were inconsistent or corrupted. This is probably caused by program overlay of library data areas.

Action

Inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

018 `ERROR: Runtime library error during dynamic loading.`
`ABEND code: 1218`

Explanation

The CMS `loadm` function failed while trying to delete a buffer. This is probably a SAS/C library error, but it may also be caused by program overlay of library data areas.

Action

Contact SAS/C Technical Support. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

019 **`ERROR: Internal error in DEBUG facility.`**
`ABEND code: 1219`

Explanation

The interface between the SAS/C library and the debugger detected an error in its own processing or in the debugger. This can occur if the program overlays debugger data areas.

Action

Contact SAS/C Technical Support. Also, inspect the program for uninitialized pointers, out-of-bounds array references and other similar errors.

---

020 **`ERROR: Program aborted by DEBUG facility.`**
`ABEND code: 1220`

Explanation

The program was terminated due to use of the debugger **`abort`** command. Also, this ABEND is issued if the debugger is unable to respond to multiple attention interrupts and the user chooses to terminate the application.

---

021 **`ERROR: System error during segment loading.`**
`ABEND code: 1221`

Explanation

An error was detected by CP while loading a library segment. This may indicate a CP system error.

Action

Contact SAS/C Technical Support for further information.

---

022 **`ERROR: Internal error in run-time domain manager.`**
`ABEND code: 1222`

Explanation

The library framework manager detected an internal error or corrupted control blocks.

Action

Contact SAS/C Technical Support.

---

023 `ERROR: System failure in TSO CLIST/Rexx interface.`
`ABEND code: 1223`

Explanation

An unexpected error occurred in the MVS SAS/C SUBCOM interface.

Action

Contact SAS/C Technical Support.

---

023 `ERROR: Required library nucleus extension dropped.`
`ABEND code: 1223`

Explanation

A CMS nucleus extension used to manage sharing of the SAS/C run-time library has been dropped by the user.

Action

Don't issue NUCXDROP for nucleus extensions whose names begin with the characters L$C.

---

024 `ERROR: Attempt to terminate non-C routine with longjmp.`
`ABEND code: 1224`

Explanation

The program issued a call to **`longjmp`** or **`exit`**, but execution of the call would require termination of one or more non-SAS/C calling routines.

Action

For some applications, allowing non-C calling routines to be terminated by **`longjmp`** or **`exit`** is not a problem. For such applications, you can modify L$UPREP to suppress this ABEND, as described in Appendix 6, "Using the indep Option for Interlanguage Communication," in the *SAS/C Compiler and Library User's Guide.*

---

025 `ERROR: Program terminated due to [name] signal.`
`ABEND code: 1225`

Explanation

A signal occurred for which the default action is termination, and no signal handler was defined. These signals include **`SIGALRM`**, **`SIGIUCV`** and most of the OpenEdition signals.

Action

If abnormal termination is not desired, modify the program to define a handler for the signal.

---

026 `ERROR: Internal error in fullscreen support library.`
`ABEND code: 1226`

Explanation

The FSSL library detected an internal error. This can occur if the application overlays FSSL data areas.

Action

Contact SAS/C Technical Support.

---

027 `ERROR: Runtime library error in coprocess support.`
`ABEND code: 1227`

Explanation

A library failure occurred terminating a coprocess.

Action

Contact SAS/C Technical Support.

---

028 `ERROR: Unable to obtain memory for IUCV interrupt handling.`
`ABEND code: 1228`

Explanation

The library was unable to allocate memory to extend the library's queue of incoming IUCV messages. This can occur if the program blocks IUCV interrupts or fails to respond to them in a timely fashion.

Action

Make sure the program has not blocked IUCV signals, and is running in an adequate virtual machine.

---

028 `ERROR: Unable to obtain memory for TCP/IP interrupt handling.`
`ABEND code: 1228`

Explanation

The library was unable to allocate memory for handling IUCV interrupts for the IBM TCP/IP product. This is probably a library internal error.

Action

Contact SAS/C Technical support.

---

029 `ERROR: No storage available to satisfy an unconditional GETMAIN request.`
`ABEND code: 1229`

Explanation

A request by a C program for more memory failed, and execution was unable to continue. This normally indicates that CICS storage was exhausted or almost exhausted.

Action

Consult with your CICS Systems Programming Staff.

---

030 `ERROR: OpenEdition not available, execution unable to continue.`
`ABEND code: 1230`

Explanation

The program invoked a function that requires the presence of OpenEdition in the system, such as **`getpid`**, but OpenEdition was not installed or not running. The POSIX definition of the function did not permit the function to fail.

Action

Do not use functions that require OpenEdition on systems where OpenEdition is not installed. If OpenEdition is normally running on your system, contact your Systems Programming Staff for assistance.

---

033 `ERROR: Internal error in interlanguage communication.`
`ABEND code: 1233`

Explanation

This message indicates an internal library error.

Action

Contact SAS/C Technical Support.

---

034 `ERROR: Inter-language call failed - no framework created for C.`
`ABEND code: 1234`

Explanation

An ILC call was attempted from a non-C environment but no valid C framework had been created.

Action

Call **`CFMWK`** before calling a C function from another language, and verify that the **`CFMWK`** call was successful. Refer to the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for details on how to create a C framework.

---

035 **`ERROR: Fatal interlanguage communication usage error.`**
`ABEND code: 1235`

Explanation

A library U1235 ABEND was issued for an unrecoverable usage error. It should always be accompanied by another message describing the nature of the error.

Action

Refer to previous messages for the cause of this error.

---

039 **`ERROR: Unexpected internal error during abnormal termination.`**
`ABEND code: 1239`

Explanation

An unexpected internal error occured in CICS ABEND handling.

Action

Contact SAS/C Technical Support.

---

040 **`ERROR: Unexpected error in ABEND analysis - analysis terminated.`**
`ABEND code: 1240`

Explanation

During ABEND analysis, an unexpected program check was encountered which made further recovery and analysis impossible. This message is produced, the analysis is terminated, and if possible a dump is taken. This error can be caused by program errors which cause library object code to be overlaid.

Action

Refer to previous messages as to the cause of the ABEND.

---

041 **`ERROR: Program terminated by operating system. ABEND code is [code].`**

Explanation

The operating system or the SAS/C library forced abnormal termination of the program. The ABEND code may be either a system code or a user code.

Action

If the ABEND code is a user code in the range 1200 to 1240, (for instance U1235) then refer to the *SAS/C Library Reference* for more details.

For a system ABEND, refer to the appropriate operating-system-dependent listing of ABEND codes for details on the ABEND. (See *MVS System Codes* in MVS or *VM System Messages and Codes* in CMS.) User ABENDs not in the SAS/C range are generated by the SAS/C program using the **`abend`** function, by assembler subroutines, or by other products, such as FORTRAN, PL/I or IMS.

Two particularly common user ABEND codes are 240, which is a FORTRAN ABEND code, and 4000, which is a PL/I ABEND code. In general, the meaning of a user ABEND code depends on the application and the other products or interfaces it uses, and therefore you should consult with local expertise if you receive a user ABEND code you do not recognize.

Optional Message Texts for 000-041

Library messages 000 through 041 are often accompanied by one or more additional messages giving supplementary information about an ABEND. These messages are not numbered. The messages and their meanings are as follows:

`C run-time storage has been overlaid by the program.`
The ABEND analysis has found one or more library control blocks to be in an inconsistent or corrupted state. Running the program again with the **`=storage`** run-time option may produce information about the overlay.

`Runtime library failure, library storage may have been overlaid.`
The ABEND occurred while a critical library routine was executing. Often this means that library storage was overlaid before the failure. Running the program again with the **`=storage`** option may produce information about the overlay.

`The ABEND occurred in another task or domain.`
The ABEND occurred in non-C code not called directly from your program. For example, you would get this message if the debugger ABENDed while debugging your program, or if a CMS ABEND occurred while a REXX function package was loaded but inactive.

`A non-C or system routine was running at the time of ABEND.`
The ABEND occurred in non-C code called from C, usually in a system routine.

`This ABEND could not be recovered due to environmental damage.`
Corruption of library control blocks prevented the library from converting this ABEND into a C signal.

`Traceback terminated – save areas overlaid.`
Due to corruption of stack information, a complete C calling trace could not be produced. If you run the program again with the **`=storage`** option, you can probably get more information about the overlay.

`Unable to determine location of failure.`
ABEND analysis was unable to determine where the failure occurred. This may be due to overlay of control blocks. It may also be generated if the ABEND occurs during program termination after the C framework has been partially destroyed. After this message, the ABEND handler prints the ABEND PSW and registers, which may be useful in the absence of a traceback.

042 **`NOTE: C run-time storage usage statistics [statistics]`**

Explanation

This message was issued due to use of the **`=usage`** run-time option. The library produces a report on the amount of stack and heap used as well as the number of system allocate requests and frees. If the program uses coprocesses, a stack report is given for both the main coprocess, and for all coprocesses combined.

043 **`NOTE: Invalid run-time argument ignored: ''=[argument]''`**

Explanation

An program argument beginning with an equals sign was specified. Arguments of this type are considered to be SAS/C run-time options; however, [argument] was not recognized as a valid option.

Action

Refer to the *SAS/C Compiler and Library User's Guide*, Chapter 8, "Run-Time Argument Processing," for information on run-time arguments.

044 **`NOTE: Further warning message will be suppressed.`**

Explanation

The library stops generating warning messages after a maximum number of five have been printed if **`stderr`** is unavailable, in order to avoid flooding MVS consoles with messages. There is no message limit if **`stderr`** is available.

Action

If possible, correct any problems that prevent the use of **`stderr`**. For instance, in MVS batch, make sure a valid SYSTERM DD statement is defined. Alternately, consider use of the **`quiet`** function to suppress these messages.

045 **`ERROR: No memory available for ABEND analysis.`**

Explanation

During ABEND analysis on MVS the ABEND handler could not obtain the memory needed to continue. The ABEND analysis and further execution were terminated.

Action

Refer to previous messages to determine the cause of the ABEND or increase memory availability and rerun the program to obtain a traceback and other diagnostic information. Note that if, due to an error, a program loops consuming all available memory, it may not be possible to bypass this condition by increasing the region size.

045 `NOTE: CP VMDUMP command issued.`

Explanation

On CMS, the SAS/C ABEND handler issued the CP VMDUMP command because the `=abdump` run-time option was specified.

046 `NOTE: Undefined items in library option string ignored.`

Explanation

When the C framework is created by the ILC routine `CFMWK`, or when the entry point to a program is specified as `$MAINO`, the caller provides an argument string containing any run-time options to be passed to the program. This diagnostic is produced if any part of this string cannot be recognized as a valid run-time option or redirection.

Action

Correct the calling program to specify a correct run-time options string. See the *SAS/C Interlanguage Communication* book for information on `CFMWK`. For more details on passing run-time arguments using `$MAINO` see the *SAS/C Compiler and Library User's Guide,* Chapter 10, "Communication with Assembler Programs."

047
```
WARNING: Run-time message texts not found. errno=[errno value]
WARNING: Run-time message texts not found, further information
unavailable.
Errno value: see message
```

Explanation

A condition occurred requiring a library message. However, the library message texts could not be accessed.

Action

If the program is executing in the normal C environment make sure the transient library is available and was installed correctly. If the program is linked with a Generalized Operating System (GOS) library, consult your local GOS implementor for information on this condition.

Note that this message may be produced by an all-resident program if a library diagnostic is required, but the program defined the symbol `NO_WARNING` before including `<resident.h>`.

048
```
NOTE: Most recent C run-time modules not available.
Use version [num] or later to avoid problems.
```

Explanation

When the C run-time environment is first created a check is made to confirm that the transient run-time library modules are at least as recent as the resident libraries used at link time. If the resident library is more recent then the transient library, this message is produced as a warning that correct behavior cannot be assured.

The second line of the message indicates the minimum transient library level necessary to guarantee compatibility with the executing load module.

Action

Contact your local support for SAS/C for information on how to access the most current version of the SAS/C transient library. For more details on combining different release levels of the resident and transient libraries, see "Rules for Using Different Releases of the Compiler and Library" in Chapter 1 of the *SAS/C Library Reference, Volume 1.*

049 `NOTE: Abbreviations must be unique.`

Explanation

This message is accompanied by message 043. An abbreviation of a run-time option was specified, but the abbreviation did not contain enough characters to identify it uniquely. The option is ignored and execution continues.

Action

Refer to the *SAS/C Compiler and Library User's Guide*, Chapter 8, "Run-Time Argument Processing" for a list of valid abbreviations.

050
```
ERROR: Execution terminated due to =QUIT run-time option.
ABEND code: various
```

Explanation

A library message was generated and the `=quit` run-time option was specified. The program is abnormally terminated, with a user ABEND code matching the number of the generated message. (For example, if message 502 was generated, then the ABEND code is user 502). This message is generated to explain the reason for the ABEND.

051 `ERROR: ABEND [code] in [language].`

Explanation

This message is generated to identify the ABEND code and the language when an ABEND occurs in a language other than C. Some languages, such as FORTRAN, modify the ABEND code during their processing, and the ABEND code printed by C may be the modified rather than the original code. [language] may also be CDEBUG for an ABEND that occurs in the C debugger or CMS for an ABEND that occurs in CMS while a REXX function package is idle.

Action

Refer to previous messages as to the cause of the ABEND. If the ABEND is in the SAS/C debugger, contact Technical Support.

052 `ERROR: [language] ABEND [code] reinstated as 0C6.`

Explanation

This message may occur when an 0Cx ABEND occurs in a language that does not define a handler for this particular ABEND. The C framework manager assumes control and determines whether or not the ABEND should proceed.

In some cases, it is not possible for the framework manager to allow the ABEND to proceed without either changing the ABEND code or the perceived location of the ABEND. Because changing the location of the ABEND affects the accuracy of the error message, the framework manager instead changes the ABEND code. More exactly, it loads the registers at the time of ABEND and branches to an odd address within a byte of the original ABEND location. Thus, information such as the number of the line that failed should be correct.

Action

Refer to previous messages as to a possible cause of the ABEND. Any messages produced by the other language will probably also be useful.

---

053 `ERROR: Inter-language call or return attempted during program termination.`

Explanation

The framework for another language had begun to terminate and then attempted to call (or return to) C. This can occur only if an interlanguage call or return is attempted after the use of a language facility that allows program termination to be intercepted, such as a PL/I FINISH ON-unit.

Action

Correct the non-C code not to call C functions during termination.

---

054 `ERROR: Internal error in C library inter-language communication routines.`

Explanation

This is a SAS/C internal library error.

Action

Contact SAS/C Technical Support.

---

055 **`ERROR: Invalid inter-language call to C.`**

Explanation

This message is issued when a C function is called from another language but the C framework is already active. The most likely cause of this error is a non-C routine that was not declared with a keyword such as **`__cobol`**. In some cases, a misdeclared routine of this sort may execute successfully; however, if it calls another C function, this error results.

Action

Refer to the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for restrictions on inter-language calls to C.

---

056 **`NOTE: SAS/C library release [`*`n.nnx`*`] (resident), release [`*`n.nnx`*`] (transient).`**

Explanation

This message indicates the versions of the resident and transient libraries for the executing program. This message is produced when the **`=version`** run-time option is used.

---

057 **`ERROR: REXX function package is not reentrant. Functions cannot be loaded.`**

Explanation

A REXX function package has been compiled using the **`norent`** compiler option. REXX function packages must be compiled using the **`rent`** option.

Action

Recompile the function package using the **`rent`** option. For complete details on REXX function packages see the *SAS/C Library Reference* chapter "The REXX SAS/C Interface."

---

058 **`ERROR: REXX environment [name] is terminating. Command rejected.`**

Explanation

This message occurs in a TSO SUBCOM application. The C program has called **`execend`** to terminate SUBCOM processing, but while the library was attempting to terminate an active TEXX EXEC, a subcommand was addressed to the terminating program.

The REXX input line that caused this message fails and passes a return code of -10 back to REXX.

Action

Correct the application so that this situation does not arise.

---

059 `NOTE: Rexx environment [name] is busy. Command rejected.`

Explanation

This message occurs in a TSO SUBCOM application. It indicates that while the C program was processing a REXX command, another command was addressed to the same environment. This is likely to happen only if the user is running several REXX EXECs as independent subtasks, or if the C program calls the **`system`** function to invoke another EXEC.

The REXX input line that generates this message fails and passes a return code of -10 back to REXX.

Action

Correct the application so that this situation does not arise.

---

059 `NOTE: SUBCOM environment is busy. Command rejected.`

Explanation

A CMS SUBCOM application invoked another CMS processor. For instance, it used the **`system`** function to invoke an EXEC. The invoked processor then attempted to address the SUBCOM environment of the original program. Recursive invocation of the SUBCOM interface in this fashion is not permitted. A return code of -10 is passed back to the program that attempted the recursive invocation.

Action

Do not attempt to address a C SUBCOM environment recursively. This may require restricting the use of general-purpose EXECs from a SUBCOM application.

---

060 `NOTE: Beginning storage corruption check.`

Explanation

When the **`=storage`** run-time option was requested, this message indicates that the storage corruption report is being generated.

---

061 `NOTE: Storage corruption report generated.`

Explanation

When the **`=storage`** run-time option was requested, this message indicates the storage corruption report has been successfully generated.

---

062 `NOTE: No storage corruption detected.`

Explanation

When the **`=storage`** run-time option was requested, this message indicates that a storage check was performed without finding any storage overlays.

---

063 `NOTE: Unable to produce storage corruption report.`

Explanation

This message indicates that for some reason the `=storage` run-time option failed to produce a storage corruption report. Typically this message is preceded by other library messages describing the problem.

Action

In MVS the most likely cause for this message is that the STGRPT DDname was not allocated.

---

064 `NOTE: ABEND diagnostic messages directed to ddname [name]`

Explanation

The library diagnostics associated with a program ABEND (including the program traceback) were directed to the DDname indicated, either SYSTERM or a SYSOUT file allocated by the library. This avoids increasing the load on the MVS system console for C programming errors.

---

065 `ERROR: Diagnostic output ABEND: [code]. Messages redirected to job log.`

Explanation

The library attempted to send diagnostic messages to a file, but an ABEND occurred while writing to the file. Use of the file is abandoned and the messages are written to the job log.

---

066 `ERROR: Unable to allocate [ddname] - DYNALLOC error code [hex].`

Explanation

An OpenEdition application attempted to dynamically allocate a DDname needed by the library as specified by an environment variable but the allocation failed. The DYNALLOC error code is included in the message.

Action

Look up the error code in the *IBM MVS/ESA Application Development Guide: Authorized Assembler Language Programs*, and take appropriate action. If the error code indicates a problem with the file to be allocated (for instance, it does not exist), correct the value of the environment variable and rerun.

---

067 `ERROR: Unable to reopen CTRANS after fork - child terminated. ABEND code: 1212`

Explanation

After a **fork** call, the child process was unable to reopen CTRANS to access the SAS/C transient library.

Action

Contact SAS/C Technical Support.

---

068 `NOTE: Error detected in file: [name]`

Explanation

An error in a run-time option or redirection was detected in the named file. The error itself is described in a previous message.

---

069 `NOTE: Terminal/pipe not valid for argument redirection.`

Explanation

An argument redirection request specified a file that was a terminal file or a pipe. These files are not supported for argument redirection.

Action

Correct the specification.

---

070 `NOTE: Enter IC to interrupt program or a REXX immediate command.`

Explanation

This message is produced if you hit the attention key while a SAS/C SUBCOM application is running a REXX EXEC. The message indicates that you can communicate to REXX by entering any REXX immediate command, or send an attention signal to the SAS/C debugger or library by entering IC.

Action

Enter a Rexx immediate command, or enter IC to signal the SAS/C application or debugger, or enter a null line to simply continue execution of the REXX EXEC.

---

071 `NOTE: Recursive inclusion of argument file: [name].`

Explanation

An argument redirection file contains another argument redirection that (directly or indirectly) causes the file to be included again. The recursive specification is ignored.

Action

Correct the file to not include a recursive specification.

---

072 **`ERROR: This version of the transient library does not support OpenEdition.`**
`ABEND code: SEC6`

Explanation

A program called using OpenEdition **`exec`** linkage determined that the accessible version of the SAS/C transient library did not support OpenEdition. The program is aborted.

Action

Provide a CTRANS DD statement or use the **`ddn_CTRANS`** environment variable to define the location of a compatible version of the transient library. Contact your SAS/C Support Representative to determine the appropriate data set name.

---

073 **`NOTE: Library internal error, text for message [number] was truncated.`**

Explanation

This message indicates that an error occurred formatting the previous library message, and that some of the message text was lost. This condition should not occur and indicates a potential problem in the run-time library.

Action

Report the problem to SAS/C Technical Support.

---

074 **`NOTE: I/O support code not accessible. Diagnostic information may be lost.`**

Explanation

This message is produced if you run an all-resident application under the shell and the library needs to generate a diagnostic message, but it is unable to send the message because HFS I/O support was not linked into the module. The original diagnostic message is sent to the MVS console as an alternative.

Action

Modify the application to define the macro **`RES_HFS_STDIO`** before including **`<resident.h>`**.

---

098 **`WARNING: VSAM PHYSICAL ERROR SYNADAF INFO:`**
`Errno value: EDEVICE`

Explanation

This message indicates a physical I/O error processing a VSAM data set, probably a hardware problem. The rest of the text of the message contains details that may be helpful in isolating the error. This message is sent to the Operator's console in MVS.

Action

Check for other indications of hardware problems. For instance, in MVS, look for IOS000I messages that may contain more information about the error.

---

099 **`ERROR: [I/O type] ERROR ([access method]) DIAGNOSTIC INFO, [info]`**
`Errno value: EDEVICE`

Explanation

This message is similar to message 098 except that it applies to BSAM rather than VSAM files. This may indicate either a hardware problem, or a program logic error. If the message is accompanied by an MVS IOS000I message, it probably indicates a hardware problem.

Action

If the message indicates "WRNG.LEN.RECORD", the probable cause is incorrect DCB information, possibly caused by incorrect concatenation of DD statements. Another possible cause is a call to **`fseek/fsetpos`** with a corrupted or uninitialized seek address. Consult with your local System Programming Staff for help interpreting this message and determining possible causes.

---

100 **`WARNING: Memory unavailable for new allocation: [num] bytes required.`**
`Errno value: ENOMEM`

Explanation

The library attempted to allocate the number of bytes indicated. This may be caused by a direct call to **`malloc`**, or may represent a library internal storage request.

Action

Increase the region size or virtual machine size. If the amount of memory requested appears unreasonable, verify that the argument passed to **`malloc`** has been correctly initialized.

---

101 **`ERROR: Program does not contain a ''main'' function. Execution aborted.`**
`ABEND code: 1210`

Explanation

A C load module was linked with neither a **`main`** function nor a **`_dynamn`** function. This message is issued by a dummy **`main`** function contained within the library. This is usually a link-edit error that caused part of the program to be omitted or that specified the wrong entry point.

Action

Unless the compiler option **`indep`** is in use, every C program requires a **`main`** function. If **`indep`** is not in use, ensure that the **`main`** function is included when the program is linked. If **`indep`** is in use, ensure that the program entry point is defined as the first function to be executed, not as MAIN.

---

102 **`WARNING: Module name is too long or entirely blank: [module name].`**
`Errno value: EARG`

Explanation

A call to **`loadm`** or **`buildm`** specified a module name that was entirely blank or contained more than 8 characters.

Action

Correct the program to make sure a valid module name is specified.

---

105 **`WARNING: Input data does not match format string: [format string].`**
`Errno value: ECONV`

Explanation

During the execution of one of the **`scanf`** family of functions, an incompatibility was detected between the data and the format string. Up to eight characters of the format string are displayed in the message.

Action

Refer to the *SAS/C Library Reference* for more details on the function being invoked and valid format strings. The **`quiet`** function can be used to suppress the message.

---

106 `WARNING: % found at end of [function] format string.`
`Errno value: EARG`

Explanation

A trailing "%" was found in a format string. The "%" denotes the beginning of a format and cannot appear by itself.

Action

If an actual "%" character is desired in the output, specify two "%" characters (for instance, "Interest rate %d%%"). Refer to the *SAS/C Library Reference* for more details on the function being invoked and valid format strings.

---

107 `WARNING: Invalid number base for strtol: [base].`
`Errno value: EDOM`

Explanation

The base specified in the invocation of **`strtol`** is invalid.

Action

Specify a valid base between 2 and 36.

---

110 `NOTE: Arguments to [function] overlap (length=[len]). Results are unpredictable.`
`Errno value: unchanged`

Explanation

The function indicated has been passed arguments that address overlapping areas. The results are unpredictable. Some of the functions that may cause this message to be generated are: **`memcpy`**, **`memcpyp`**, and **`memfil`**. This message may also be generated if the length passed to one of these functions is extremely large, perhaps due to an uninitialized variable.

Action

If it is valid for the data areas to overlap, consider using the **`memmove`** function. See the *SAS/C Library Reference* for information on **`memmove`**.

This message can be generated within the library during program startup if a SAS/C program is invoked with an invalid parameter list pointer in register 1. If the program issuing the mesage was invoked from another program (for instance, using the ATTACH macro), check the invoking program to be sure it has constructed a correct parameter list.

---

111 **`NOTE: Length of memset target too large: [num]. Results unpredictable.`**
`Errno value: unchanged`

Explanation

The program is running in AMODE 24, and a call to **`memset`** specified a length of more than 16 megabytes. The probable cause is an uninitialized argument to **`memset`**.

Action

Correct the program.

---

114 **`WARNING: [error] occurred converting ``[string]'' to floating point.`**
`Errno value: ERANGE`

Explanation

During an attempt to convert a string to a floating point number, an error occurred. The error is either "Underflow" or "Overflow."

Action

None. The data to be converted represented a number outside the hardware range for floating-point numbers.

---

118 **`WARNING: buildm unsuccessful. Module name [name] already defined.`**
`Errno value: EUSAGE`

Explanation

The module name specified to **`buildm`** is the name of a previously loaded or built module.

Action

Each module name should normally be unique. If reusing an existing module name is required, refer to the *SAS/C Library Reference* description of **`buildm`** for details.

---

119 **`WARNING: Transient module [module name] could not be located.`**
`Errno value: EFORBID`

Explanation

An all-resident program required a transient load module, but the symbol **`ALLOW_TRANSIENT`** was not defined when it was compiled.

Action

This generally indicates that the program has attempted to use a library feature which was not specified at the time that the **`<resident.h>`** header file was included. This is permitted only if **`ALLOW_TRANSIENT`** is specified and the transient library is accessible. Refer to the *SAS/C Compiler and Library User's Guide*, Chapter 9, "All Resident C Programs," for details.

---

121 `WARNING: Calendar time value outside the range of the system clock.`
`Errno value: EUSAGE`

Explanation

The IBM 370 system clock has a year range of 1900-2041. If the time value passed to the **`mktime`** function falls outside of this range, the above message is generated and **`mktime`** returns a error code to its caller.

Action

Avoid the use of time values outside the valid range.

---

122 `WARNING: Minimum field width of 0 ignored in [function] format.`
`Errno value: EUSAGE`

Explanation

A field width of 0 was specified in the format for a **`scanf`** type function. This width is illegal and execution of **`scanf`** is terminated.

Action

Specify a field width other than 0.

---

124 `ERROR: Assertion failed: [assertion].`
`ABEND code: 1210`

Explanation

An assertion was specified in the source program and the assertion was not met. The executing program is terminated (**`abort`** is called).

Action

If the assertion is incorrect, remove or correct it. If the assertion is correct, fix the program logic error which caused the assertion to fail.

---

125 `WARNING: Invalid [function] format character '[char]'.`
`Errno value: EARG`

Explanation

An invalid format specification such as "**`%q`**" was passed to a **`printf`** or **`scanf`** type function.

Action

For more details on valid formats see the *SAS/C Library Reference* description of the specified function.

---

126 **`WARNING: [hex] is not a valid double byte character.`**
`Errno value: EARG`

Explanation

The hex characters printed do not comprise a valid double byte character.

Action

For more details see the *SAS/C Library Reference*, Chapter 11, "Multibyte Character Functions."

---

127 **`NOTE: Unmatched double byte shift-out in [function] argument string.`**

`Errno value: EARG`

Explanation

A zero byte was encountered within a double byte character sequence, so the sequence was terminated.

Action

For more details see Chapter 11, "Multibyte Character Functions," in *SAS/C Library Reference, Volume 1.*

---

128 **`WARNING: %[format code] conversion exceeds maximum length of [num] characters.`**
`Errno value: EARG`

Explanation

A format item for a **`printf`**-like function specified a length larger than the limit for this function. The limit is 512 characters for functions other than **`sprintf`** and its variants (which have no limit). Note, this limit does not apply to "**`%s`**" and "**`%V`**" formats.

Action

Decrease the size of the conversion, or use **`sprintf`** or **`snprintf`** to format the data, followed by a function such as **`fputs`** to write the formatted output.

---

129 **`WARNING: Integer overflow converting ``[string]'' to long (base [num]).`**
`Errno value: ERANGE`

Explanation

During an attempt to convert the specified string to the specified base, the **`strtol`** function encountered an integer overflow. The **`strtol`** function returns **`LONG_MAX`** or **`LONG_MIN`** for this condition. **`stroul`** can also generate this message, in which case it returns **`ULONG_MAX`**.

Action

A possible solution is to use **stroul** (unsigned long) for the conversion. Refer to the *SAS/C Library Reference* description of **strtol** and **strtoul** for more information.

130 **ERROR: Attempt to free an address not allocated by malloc.**
Abend code: 1208

Explanation

The argument of a call to **free** did not appear to be the address of storage previously allocated by **malloc**. One possible explanation of this condition is that the 8-byte prefix which precedes each block of storage allocated by **malloc** has been overlaid. Note that the overlay may have occurred much earlier in the program's execution and may have nothing to do with the code executing at the time of the ABEND.

Action

If the storage to be freed appears to have been correctly allocated with **malloc**, run the program again using the **=storage** run-time option, or run the program under the SAS/C debugger and use the **storage** debugger command to locate the overlaid data areas. Also, inspect the program for uninitialized pointers, out-of-bounds array references, and other similar errors.

131 **WARNING: Invalid signal number: [num]**
Errno value: EARG

Explanation

A signal-related function was called specifying an undefined signal number.

Action

Ensure that the library passes valid signal names in calls to signal-handling functions. Integer constants should not be passed to signal routines, since different systems often use different numbers for the same signal. See Chapter 5, "Signal-Handling Functions" in *SAS/C Library Reference, Volume 2*, for information on valid signals.

132 **WARNING: Signal [signal] cannot be ignored.**
Errno value: EARG

Explanation

A request was made to ignore the indicated signal; however, certain signals, such as **SIGABND**, cannot be ignored.

Action

See "Signal-Handling Functions" in *SAS/C Library Reference, Volume 2*, for information on valid signals.

133 **`WARNING: Signal number already defined: [num].`**
`Errno value: EARG/EUSAGE`

Explanation

An attempt was made to define a user signal that is not in the valid range for user-defined signals or which has already been defined as a user signal.

Action

See "Signal-Handling Functions" in *SAS/C Library Reference, Volume 2*, for information on defining user signals.

---

134 **`NOTE: Handling of [signal] not permitted due to =[option] option.`**
`Errno value: unchanged`

Explanation

The named signal cannot be handled because a run-time option was used. The option **`=nohtsig`** inhibits handling of termination signals **`SIGABRT`** and **`SIGABND`**. The option **`=nohcsig`** inhibits handling of computational signals **`SIGFPE`**, **`SIGSEGV`** and **`SIGILL`**. The request to handle the signal is ignored.

Action

See the SAS/C Compiler and Library User's Guide for complete details on run-time library options and "Signal-Handling Functions" in *SAS/C Library Reference, Volume 2* for general information on signal-handling.

---

135 **`ERROR: Return from [signal] handler not supported, execution terminated.`**
`ABEND code: signal-dependent`

Explanation

A handler attempted to return to the point of interruption, but this was not allowed for this signal. The program is abnormally terminated, using the ABEND code associated with the signal. (Signals for which return is forbidden include **`SIGABRT`**, **`SIGABND`**, **`SIGSEGV`** and **`SIGILL`**.)

Action

To recover from one of these signals, the handler must use the **`longjmp`** function to resume execution. See "Signal-Handling Functions" in *SAS/C Library Reference, Volume 2* for general information on signal handling.

---

136 `ERROR: Argument to unloadm is not a valid dynamically loaded function address.`
`ABEND code: 1211`

Explanation

The program called the **unloadm** function, but the argument did not appear to be a pointer to a function in a load module loaded by **loadm**. Overlay of the **unloadm** argument can cause this message.

Action

If the argument to **unloadm** appears to be a valid function pointer to a module loaded by **loadm**, inspect the program for uninitialized pointers, out-of-bounds array references, and other similar errors. Use of the debugger **monitor** command may also be helpful in locating the overlay.

---

137 `WARNING: Attempt to allocate from uninitialized or corrupted storage pool.`
`Errno value: EUSAGE`

Explanation

The control information in a **pool_t** object is incorrect. Either the object has not been initialized by a call to the **pool** function or it has been overlaid.

Action

Check that the **pool_t** object was correctly initialized. If so, use the debugger **monitor** command to determine where the object was overlaid.

---

138 `WARNING: Insufficient memory to load module [module name].`
`Errno value: ENOMEM`

Explanation

An attempt to dynamically load the named module failed because there was not enough memory for the library to allocate its load module management control blocks.

Action

Increase the amount of memory available for the execution and make sure that the program is freeing memory which it no longer requires.

---

139 **`WARNING: buildm failed due to insufficient memory.`**
`Errno value: ENOMEM`

Explanation

The **`buildm`** routine attempted to build an entry point address as requested. During the attempt, **`buildm`** could not obtain enough memory to complete its processing.

Action

Increase the amount of memory available for the execution and make sure that the program is freeing memory which it no longer requires.

---

140 **`WARNING: Argument to coproc is unrecognized: [num].`**
`Errno value: EARG`

Explanation

The **`coproc`** function accepts only the symbolic arguments MAIN, SELF, or CALLER.

Action

See the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions," for information on **`coproc`**.

---

141 **`WARNING: Argument to [function] does not identify an active coprocess.`**
`Errno value: EUSAGE`

Explanation

An argument to the specified function is not the id of any active coprocess. Most likely, the coprocess id is uninitialized or corrupted.

Action

Check to be sure the coprocess id was correctly initialized. If so, use the debugger **`monitor`** command to determine where the id was overlaid.

---

142 **`WARNING: coreturn not permitted from main coprocess.`**
`Errno value: EUSAGE`

Explanation

The main coprocess cannot call **`coreturn`** because there is no calling coprocess to return to.

Action

Modify the program so that the main coprocess does not call **`coreturn`**. For complete details on coprocesses, see the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions."

---

143 `WARNING: Recursive cocall not permitted.`
`Errno value: EUSAGE`

Explanation

A coprocess attempted to cocall itself either directly or via one or more additional coprocesses. This behavior is not supported. The erroneous call to **`cocall`** returned and execution continued.

Action

Modify the program's logic to avoid recursive cocalls. For complete details on coprocesses, see the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions."

144 `WARNING: Maximum number of coprocesses already active.`
`Errno value: ELIMIT`

Explanation

A request was made to create another coprocess, but there were already 65535 coprocesses active.

Action

Reorganize the program so that this limit is not reached. (Note that in most cases a program which attempts to create the maximum number of coprocesses will run out of memory first.) For complete details on coprocesses, see the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions."

145 `WARNING: No memory available to start new coprocess.`
`Errno value: ENOMEM`

Explanation

An attempt to create a new coprocess failed due to insufficient memory.

Action

Increase the amount of memory available for the execution and make sure that the program is freeing memory which it no longer requires. Also, investigate whether the amount of stack space for each coprocess can be reduced.

146 `WARNING: Unable to start new coprocess during program termination.`
`Errno value: EUSAGE`

Explanation

An attempt was made to start a new coprocess while a program was terminating, for instance, while an **`atexit`** routine was running. This is not permitted.

Action

Create any necessary coprocesses before termination. For complete details on coprocesses, see the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions."

147 **`ERROR: Unable to add atcoexit routine for terminating coprocess.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to add an **`atcoexit`** routine for a terminating coprocess. Once a coprocess has begun to terminate, it is not possible to define **`atcoexit`** routines for it.

Action

Define any necessary **`atcoexit`** routines before termination of the coprocess. For complete details on coprocesses, see the *SAS/C Library Reference*, Chapter 9, "Coprocessing Functions."

---

148 **`ERROR: Unable to add atexit routine during program termination.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to define an **`atexit`** routine while the program was terminating. This is not permitted.

Action

Define any necessary **`atexit`** routines before termination. See the *SAS/C Library Reference, Volume 1* for information on **`atexit`**.

---

149 **`NOTE: coreturn not permitted, calling coprocess no longer active.`**
`Errno value: unchanged`

Explanation

An attempt was made to return to a coprocess that is no longer active. This can happen only during program termination if a coprocess intercepts its own termination using **`blkjmp`** or **`atcoexit`**.

Action

For complete details on coprocesses, see the *SAS/C Library Reference* chapter "Coprocessing Functions."

---

150 **`WARNING: Reserved fields in costart parms not zero.`**
`Errno value: EARG`

Explanation

The **`costart_parms`** argument to the **`costart`** function contains several reserved areas that must be set to zeros. This condition probably occurred because the **`costart_parms`** argument was not completely initialized.

Action

Check that the **`costart_parms`** argument has been correctly initialized.

---

160 `WARNING: Invalid first argument to sigprocmask: [num]`
`Errno value: EARG`

Explanation

The first argument to **sigprocmask** is not one of the symbolic constants **SIG_BLOCK**, **SIG_UNBLOCK**, or **SIG_SETMASK**.

Action

Correct the program to pass a valid first argument. See Chapter 5, "Signal-Handling Functions," in the *SAS/C Library Reference* for further information on **sigprocmask**.

161 `WARNING: Signal [signal] cannot be handled.`
`Errno value: EINVAL`

Explanation

OpenEdition does not allow the indicated signal to be handled.

165 `WARNING: Daylight savings time information is missing.`
`Errno value: EUSAGE`

Explanation

A call to **mktime** specified a **tm_isdst** field of 1 in the argument, indicating that daylight savings time was in effect. But the value of the **TZ** environment variable did not include a daylight savings time definition, so the call to **mktime** failed.

Action

Either modify the **TZ** environment variable to include a daylight savings time definition, or set the **tm_isdst** field to 0 or -1.

166 `NOTE: TZ not set - tm_isdst setting cannot be used.`
`Errno value: unchanged`

Explanation

A call to **mktime** specified a **tm_isdst** field of 1 or 0 in the argument, indicating that daylight savings time was or was not in effect. Because the **TZ** environment variable was undefined or invalid, the library has no daylight savings time information available. The returned **tm_isdst** field is set to -1, and **mktime** assumes an offset from GMT based on the hardware time-of-day clock.

Action

Modify the program to specify a **tm_isdst** field of -1 if it may be run in an environment in which the **TZ** environment variable is not defined, or use the **quiet** function to suppress the message.

167 **WARNING: tm_isdst setting results in an ''impossible'' time.**
Errno value: EUSAGE

Explanation

The argument to **mktime** specified a time which cannot occur in the time zone defined by the **TZ** environment variable and set the **tm_isdst** field of its argument to -1. For instance, if standard time ends at 2:00 AM on March 30, and daylight time begins at that moment at 3:00 AM, then 2:30 a.m. on March 30 is an "impossible" time.

Action

Do not specify **mktime** arguments which are not valid times. Alternately, use the **quiet** function to suppress the message.

---

168 **WARNING: Inconsistent TZ value supplied. TZ setting will be ignored until TZ is reset.**
Errno value: EUSAGE

Explanation

The value of the **TZ** environment variable specified a time zone that is inconsistent. For example, the end time for daylight savings time is later than the end time for standard time but less than the start time for savings time. The inconsistent **TZ** value is ignored, and the time zone implied by the hardware time-of-day clock setting is used.

Action

Make sure the **TZ** environment variable setting is correct.

---

169 **WARNING: Invalid/unsupported format of TZ environment variable ''[value]''.**
Errno value: EUSAGE

Explanation

The value of the **TZ** environment variable does not conform to the syntax specified by the POSIX standard. Note that the POSIX standard allows an implementation to support additional **TZ** formats beginning with a ":" but SAS/C supports no such formats.

Action

Make sure the **TZ** environment variable setting is correct. See Chapter 3, "Function Categories," in *SAS/C Library Reference, Volume 1* for information on the form of the **TZ** environment variable.

---

170 **`WARNING: Invalid or missing environment variable name.`**
`Errno value: EARG`

Explanation

An invalid environment variable name was specified in a call to **`putenv`** or **`setenv`**. For instance, the variable contains an equals sign.

Action

Correct the environment variable name. For more information on the syntax for environment variable names, see Chapter 4, "Environment Variables," in the *SAS/C Library Reference.*

---

171 **`WARNING: Scope specified for environment variable is invalid: [scope]`**
`Errno value: EUSAGE/EFORBID`

Explanation

The scope specifed is invalid or not supported in the current environment. See the "Environment Variables" chapter of the *SAS/C Library Reference* for a discussion of the valid scopes. If **`errno`** is set to **`EFORBID`**, it indicates that the program was created using the all-resident library and that use of TSO environment variables was not requested at compile-time. It also indicates that the program was running in TSO, called **`getenv`**, and was unable to find a variable in the program scope variable list.

Action

For more information on environment variable scopes, see Chapter 4, "Environment Variables," in the *SAS/C Library Reference.* See the *SAS/C Compiler and Library User's Guide* for information on using the all-resident library.

---

175 **`WARNING: [name] locale is not supported - unable to load [name] module.`**
`Errno value: ENFOUND`

Explanation

The argument to **`setlocale`** requested the use of a non-standard locale, but no load module supporting that locale could be loaded.

Action

If the locale specified is intended to be a user-defined locale, correct JCL allocations or disk accesses so the load module supporting that locale is accessible. If the locale is intended to be a standard locale, correct the specification. See "Localization Functions" in the *SAS/C Library Reference, Volume 2* for more information on implementing user-defined locales.

---

176 `WARNING: Unrecognized text in locale string: [text]`
`Errno value: ENFOUND`

Explanation

The locale specification contained unrecognized text. A SAS/C locale consists of a single name, optionally followed by a semi-colon and a list of category overrides, separated by semicolons.

Action

Correct the locale specification. See "Localization Functions" in the *SAS/C Library Reference, Volume 2* for more information on the syntax of valid locale specifications.

---

177 `WARNING: Unrecognized setlocale category: [num]`
`Errno value: ENFOUND`

Explanation

The first argument to the **`setlocale`** function is not one of the locale category symbolic names defined in **`<locale.h>`**.

Action

Correct the program. See "Localization Functions" in the *SAS/C Library Reference, Volume 2* for further information on locale categories.

---

180 `NOTE: DEBUG module not loaded, =DEBUG option ignored`

Explanation

The **`=debug`** run-time option was specified, but the debugger load module could not be loaded. The cause of the failure should have been described by a previous message.

Action

Correct the condition described in the previous message.

---

181 `NOTE: =DEBUG option cancelled, unable to open debug processor files`

Explanation

In MVS batch, the debugger requires DBGIN (input) and DBGLOG (output) files to be available upon initialization. DD statements for these files must be present in the JCL.

Action

Correct the JCL.

---

182 `WARNING: [function] routine not called, I/O suppressed for this program.`
`Errno value: EFORBID`

Explanation

An attempt was made to perform I/O in a program for which I/O support was suppressed. I/O can be suppressed using the _NIO library option or, in an all-resident program, by specifying NO_IO when including `<resident.h>`.

Action

Correct the program to not attempt I/O, or modify it so I/O is permitted.

---

183 `NOTE: =DEBUG option not supported for authorized program.`

Explanation

The debugger is not supported for programs running in an authorized environment. If this were permitted, debugger facilities could allow subversion of MVS integrity and security features by unauthorized users.

Action

Contact SAS/C Technical Support for further information.

---

184 `NOTE: Unsupported remote debugger _DB_COMM method [text]`

Explanation

The remote communication access method specified in the _DB_COMM environment variable for either the remote debugger or program communcication is not supported.

Action

Change the _DB_COMM environment variable specification to a supported value. Supported access methods that can be specified are: TCPIP (TCP/IP access method), APPC (IBM's Advanced Program to Program Communcation), or LOCAL (non-remote debugger).

---

185 `NOTE: =DEBUG option not supported for setuid/setgid program.`

Explanation

It is not yet clear if this message will be possible with the remote debugger. The debugger is not supported for programs running with an effective uid or gid different from the real uid or gid. If this were permitted, debugger facilities could allow subversion of OpenEdition integrity and security features by unauthorized users.

Action

Contact SAS/C Technical Support for further information.

---

186 **`NOTE: The local debugger is not supported for exec-linkage programs.`**

Explanation

The use of the local debugger (**`=debug`** runtime option) is not supported for programs invoked with exec-linkage, such as applications running under the OpenEdition shell.

Action

Debug exec-linkage applications using the **`sascdbg`** shell command or TSO command.

---

200 **`WARNING: [error] occured during computation of [expression].`**
`Errno value: ERANGE`

Explanation

Floating-point overflow or underflow (as indicated in the message) occurred during computation of a mathematical function. **`errno`** is set to **`ERANGE`** and an appropriate value **`±HUGE_VAL`** for overflow or 0.0 for underflow is returned.

Action

If possible, correct the program or the data. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

---

201 **`WARNING: [error] occurred during computation of [expression].`**
`Errno value: ERANGE`

Explanation

Floating-point overflow or underflow (as indicated in the message) occurred during computation of a mathematical function. **`errno`** is set to **`ERANGE`** and an appropriate value **`±HUGE_VAL`** for overflow or 0.0 for underflow is returned.

Action

If possible, correct the program or the data. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

---

202 **`WARNING: [value] is a singularity of the [function] function.`**
`Evaluation impossible.`
`Errno value: ERANGE`

Explanation

The indicated value is a singularity of the indicated function; that is, the function approaches infinity as the argument approaches the singularity. **`errno`** is set to **`ERANGE`**, and an appropriate value of **`±HUGE_VAL`** is returned.

Action

If possible, correct the program or the data. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

---

203 **`WARNING: [value] is not a valid argument to the [function] function.`**
`Errno value: EDOM`

Explanation

The indicated function is mathematically undefined for the given argument. **`errno`** is set to **`EDOM`**.

Action

If possible, correct the program or the data. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

204 **`WARNING: [expression] is not mathematically meaningful.`**
`Errno value: EDOM`

Explanation

The indicated function is mathematically undefined for the given arguments. **`errno`** is set to **`EDOM`**.

Action

If possible, correct the program or the data. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

205 **`NOTE: [Total/Partial] loss of significance occurred during evaluation of [expression].`**
`Errno value: unchanged`

Explanation

The function result is less accurate than the arguments. If the message indicates total loss of significance, the result may be completely inaccurate. If it indicates partial loss of significance, the result is approximately correct, but considerably less precise than its input arguments.

Action

If possible, correct the program or the data. Output of this message may indicate serious problems with the correctness of program results, which should be evaluated carefully. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

206 **`WARNING: [error] occured during computation of [expression].`**
`Errno value: ERANGE`

Explanation

Floating-point overflow or underflow (as indicated in the message) occurred during computation of a mathematical function. **`errno`** is set to **`ERANGE`** and an appropriate value ± **`HUGE_VAL`** for overflow or 0.0 for underflow is returned.

Action

If possible, correct the program or the data. The message can be suppressed using the **`quiet`** function, as described in the *SAS/C Library Reference, Volume 1.*

207 **`NOTE: [Total/Partial] loss of significance occurred during evaluation of [expression].`**
`Errno value: unchanged`

Explanation

The function result is less accurate than the arguments. If the message indicates total loss of significance, the result may be completely inaccurate. If it indicates partial loss of significance, the result is approximately correct, but considerably less precise than its input arguments.

Action

If possible, correct the program or the data. Output of this message may indicate serious problems with the correctness of program results, which should be evaluated carefully. The message can be suppressed using the **`_matherr`** function, as described in the *SAS/C Library Reference, Volume 1.*

270 **`WARNING: The language [language] is not defined to the C library.`**
`Errno value: EARG`

Explanation

A call to **`mkfmwk`** or **`CFMWK`** specified a language that was unknown to the library.

Action

Check the spelling of the language. (For instance, specify “PLI”, not “PL1”.) Refer to the *SAS/C Compiler Interlanguage Communication Feature User’s Guide* for a list of valid languages.

271 **`WARNING: C framework initialization failed due to lack of memory.`**

Explanation

A call to **`CFMWK`** failed because there was insufficient memory to initialize the C framework.

Action

Make sure the program is not incorrectly allocating memory. If it is not, then increase the amount of available memory for execution.

272 **`ERROR: Inter-language call failed - target language not defined.`**
`ABEND code: 1235`

Explanation

A call was made to a routine declared with the **__foreign** keyword but more than one user-supported language was active. No "language token" was passed to define the correct language.

Action

Correct the program by having only a single "foreign" language active or by supplying a "language token" as needed. Refer to the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for more details.

---

273 **ERROR: Unexpected termination of language framework for [language].**
ABEND code: 1233

Explanation

This message indicates an internal library error.

Action

Contact SAS/C Technical Support.

---

274 **ERROR: Unsupported control structure used in multi-language program.**

ABEND code: 1235

Explanation

An interlanguage call or return has occurred but the routine making the call or return is not consistent with the chain of routines that were active at the time of the last interlanguage call or return. This could be caused by a storage overlay.

However, it is more likely that use of a GOTO-like control structure in one language (such as the C **setjmp**) has terminated routines in another language. This condition can usually only be detected at the time of an interlanguage call or return; therefore, the actual error may have occurred much earlier than the point where the message was issued.

Action

Check the program for misuse of C **longjmp**, PL/I GOTO, etc. Also, running with the C run-time option **=storage** may produce useful information if a storage overlay has occurred.

---

275 **`WARNING: Unable to create language framework for [language].`**
`Errno value: ESYS`

Explanation

The framework for the named language failed to initialize.

Action

The other language's run-time library should have generated one or more messages describing the condition that prevented initialization. Use these messages to determine how to correct the problem.

---

276 **`WARNING: Argument to dlfmwk is not a valid language token.`**
`Errno value: EUSAGE`

Explanation

An argument was passed to **`dlfmwk`** that is not a valid language token for any active language. This may be the result of an uninitialized variable. It can also occur if an attempt is made to delete a framework more times that it has been created. **`dlfmwk`** returns a nonzero value to indicate the call has failed.

Action

Check the program for one of the errors described above. See the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for more information on **`dlfmwk`**.

---

277 **`WARNING: Unable to terminate execution of [language].`**
`Errno value: EUSAGE`

Explanation

A call **`dlfmwk`** or **`DCFMWK`** failed because the framework could not be terminated. One or more routines in the named language may have been still executing. For example, this message is generated if a C function called from FORTRAN uses **`dlfmwk`** to attempt to delete the FORTRAN framework. A nonzero value will be returned to indicate the call has failed.

Action

Check the program for unterminated routines in the target language. For a call to **`DCFMWK`**, specify the **`=btrace`** run-time option to get a trace of still active routines.

---

278 **`WARNING/ERROR: Inter-language communication not permitted for more than one coprocess.`**
`Errno value: EUSAGE`
`ABEND code: 1235`

Explanation

An attempt was made to create a framework or call a routine in another language in more than one coprocess. If the error is detected by **`mkfmwk`**, the message is issued as a WARNING and a zero token is returned to indicate that the framework could not be created.

If the error is detected during a call to another language, the message is issued as an ERROR and a user 1235 ABEND is issued because there is no way to communicate to the program that the call has failed.

Action

Refer to the *SAS/C Compiler Interlanguage Communication Feature User's Guide* for more information.

---

279 `ERROR: Inter-language call failed - no framework created for [language].`
`ABEND code: 1234`

Explanation

An attempt was made to call a routine in another language when its framework had not yet been successfully created. A call to **`mkfmwk`** or **`CFMWK`** may have been omitted.

Alternately, perhaps **`mkfmwk`** or **`CFMWK`** failed and the program neglected to check for failure, in which case a previous message should describe the reason for the failure. After this message is printed, the SAS/C library issues a user 1234 ABEND.

Action

Correct the program to successfully create a language framework before calling a routine in that language.

---

280 `WARNING: Attempt to terminate language frameworks out of order. Errno value: EUSAGE`

Explanation

Calls to **`dlfmwk`** or **`DCFMWK`** were made in an incorrect order. The language whose framework was created last must always be deleted first. A nonzero value was returned to indicate the call has failed.

Action

Correct the program to terminate frameworks in the correct order.

---

281 `NOTE: [language] framework terminated unexpectedly - attempting to halt execution.`

Explanation

The named language terminated while other frameworks were still active. There are many possible causes for this message, including the following:

- A routine in the named language terminated as the result of executing a termination request, such as a C **`exit`** call or a FORTRAN STOP statement.
- The main routine completed execution without successfully deleting all of the frameworks it created.
- The run-time library for the named language terminated the framework due to an error detected by it, such as a PL/I ON-condition.
- The framework was terminated as the result of debugging. For instance, the SAS/C debugger's **`exit`** command was used.

The SAS/C library responded to this condition by terminating all frameworks and program execution.

Action

If program termination was not intended, correct the program. For instance, perhaps a FORTRAN routine used a STOP statement when RETURN was intended, or a COBOL program used a STOP RUN when GOBACK was intended.

---

282 **`ERROR: Inter-language call or return attempted during program/process termination.`**
`ABEND code: 1235`

Explanation

The C program had begun to terminate and then attempted to call (or return to) another language. This can occur only if an interlanguage call or return is attempted after **`blkjmp`** or **`atexit`** is used to intercept program termination. This message can also occur if an interlanguage call is attempted during termination of a coprocess using **`blkjmp`** or **`atcoexit`**.

---

283 **`NOTE: Additional errors may occur due to [language]'s premature termination.`**

Explanation

This message is generated after message 281 if the framework that terminated unexpectedly is not the framework most recently created. In this case, the unexpected termination interfered with the SAS/C framework manager's error-handling code. This interference is only temporary; if termination completes successfully, there should be no residual effects of the interference.

However, if an error or program check occurs during termination of the other frameworks, this will likely provoke a cascade of errors, with little reliable information about cause or location.

Since the possibility of errors during framework termination cannot be prevented, this message is issued before anything else can go wrong because the chances of issuing it after an error are slim.

Action

None required.

---

284 **`NOTE: Argument [num] to [language] routine is not a supported data type.`**

Explanation

An argument passed to a non-C routine is of a data type not supported by the called language. For instance, a structure was passed to FORTRAN. The SAS/C library passes the argument by reference, unaltered.

Action

If the call is correct, you can inhibit this message by using an & or @ operator in the argument to pass it as a pointer.

---

285 **`NOTE: Return value from [language] routine is not a supported data type.`**

Explanation

The declared return value type for a non-C routine called from C is not a type supported by the called language. For instance, a PL/I routine cannot return a structure. The SAS/C library allows the call to proceed but the handling of the return value is unpredictable.

Action

If the results are not satisfactory, rewrite the interface between the languages to use a supported data type. The message can be suppressed via the **`quiet`** function, as described in the *SAS/C Library Reference, Volume 1.*

---

286 **`WARNING: Unable to create framework. Too many frameworks currently active.`**
`Errno value: ELIMIT`

Explanation

A library table used to contain framework information has filled up and no more frameworks can be created. This table is shared by all C programs in a virtual machine or under a single MVS task. The table contains entries for 20 frameworks. **`CFMWK`** stores a zero token value to indicate the failure.

Action

You can normally correct this condition by running fewer C programs simultaneously in a single virtual machine or task.

---

287 **`NOTE: Calling coprocess terminated unexpectedly - attempting to halt other languages.`**
`Errno value: unchanged`

Explanation

A coprocess (other than the main coprocess) that has created frameworks for additional languages terminated without deleting these frameworks. These frameworks are all deleted, but execution of the C program continues.

Action

No response is necessary; however, it is good practice to have the program delete the frameworks.

---

288 **NOTE: [language] framework terminated unexpectedly - attempting to halt calling process.**
Errno value: unchanged

Explanation

A language, whose framework was created by a C coprocess other than the main coprocess, terminated unexpectedly. All other non-C frameworks, and the coprocess that created them, are terminated, but other coprocesses continue to execute if there are no errors during this termination. See message 281 for a list of possible causes of unexpected framework termination.

Action

Check to be sure that termination of the non-C framework was intentional. See the Action for message 281.

---

289 **ERROR: Inter-language call failed due to previous error - errno=[code]**
ABEND code: 1235

Explanation

An unexpected error occurred during a call from C to a non-C routine. Previous message(s) should describe this error more fully. The [code] value in the message is the value stored in **errno** by the previous error. Usually this value will be ENOMEM, indicating that there was no memory available to perform the call.

After this message is printed, the library issues a user 1235 ABEND.

---

300 **WARNING: File not opened, print=yes cannot be specified for an update file.**
Errno value: EUSAGE

Explanation

Simultaneous reading and writing of a **print=yes** file is not supported. Note, a file is **print=yes** by default if it has A in its RECFM and is opened in text mode.

Action

Correct the program. One possible workaround would be to modify the program to use binary I/O to manipulate the carriage control information in column 1 of the data records directly.

---

301 **WARNING: Keyword ''[name]'' cannot be specified for [type] file access.**
Errno value: EUSAGE

Explanation

The amparm keyword specified is not valid for the type of file being opened. Examples of invalid combinations are:

Keyed access:

```
print, page, pad, eof, prompt, trunc, commit
```

Binary access:

```
print, page
```

---

302 `WARNING: File unusable due to previous error.`
`Errno value: EPREV`

Explanation

A error occurred in a previous I/O request and the file's error flag has not been cleared (see **clearerr** and **clrerr**). The library cannot process this file further.

Action

Refer to the previous messages for more details. If you wish to continue using a file after an error, call the **clearerr** or **clrerr** function, as described in the *SAS/C Library Reference, Volume 1.*

---

303 `WARNING: Attempt to write read-only file.`
`Errno value: EUNSUPP`

Explanation

This is a library internal error.

Action

Report the problem to SAS/C Technical Support.

---

304 `WARNING: File does not support seeking.`
`Errno value: EUNSUPP`

Explanation

An attempt was made to seek using **fseek**. Certain file types do not support seeking, such as the terminal and SYSOUT files.

Action

See Tables 3.5 and 3.6 in *SAS/C Library Reference, Volume 1*, for a list of file types for which seeking is not supported.

---

306 `WARNING: Attempt to write short record to format F file with pad=no.`

`Errno value: EUNSUPP`

Explanation

An attempt was made to write out a partial record. A file with a fixed record format, RECFM=F, requires all records to be full-length. The library will only pad short records if the **pad** amparm is specified. Valid **pad** specifications are **pad=null** and **pad=blank**. Note, for binary files accessed using the **''seq''** access method, **pad=no** is assumed by default.

Action

Either correct the program to write full-length F format records, change the output RECFM, or specify an appropriate **pad** amparm when opening the file. Refer to "I/O Functions" in Chapter 3 of the *SAS/C Library Reference, Volume 1.*

---

307 **`WARNING: Invalid final argument to [function].`**
`Errno value: EARG`

Explanation

An invalid seek type was passed to the indicated function. The functions **`fseek, lseek and kseek`** only support seek types of **`SEEK_SET, SEEK_CUR,`** and **`SEEK_END`**.

Action

Correct the erroneous call to specify a valid symbolic constant as the final argument. Refer to "I/O Functions" in Chapter 3 of the *SAS/C Library Reference, Volume 1* for descriptions of specific functions.

---

309 **`WARNING: The terminal cannot be opened for update.`**
`Errno value: EFATTR`

Explanation

Opening the terminal for both reading and writing is not supported by SAS/C.

Action

To perform both input and output to the terminal, open them separately. To assume complete control of the terminal, use the SAS/C full-screen support library and window manager. For more details refer to *SAS/C Full-Screen Support Library User's Guide, Fourth Edition.*

---

310 **`WARNING: Improper file mode for open: [text]`**
`Errno value: EARG`

Explanation

The open mode passed to one of the **`fopen`** family of functions is invalid.

Action

Refer to "I/O Functions" in Chapter 3 of the *SAS/C Library Reference, Volume 1*, for a detailed description of the permitted open modes.

---

311 **`WARNING: Access method ''[name]'' not defined.`**
`Errno value: EARG`

Explanation

The access method passed to **`afopen`** is not defined. The access method name must be no more than four characters in length. Valid access methods are: **`''term''`**, **`''seq''`**, **`''rel''`**, **`''kvs''`**, and **`''hfs''`**.

This error may occur with an all-resident program if an unusual type of file is processed and appropriate symbols were not defined in the source module that includes **`<resident.h>`**. For example, you might get this message for access method **`''kvs''`** if **`RES_VSAM`** was not defined and the program attempted to access a VSAM file.

Action

Refer to the section "Library access methods" in Chapter 3 of the *SAS/C Library Reference* for more detail. See "All-Resident C Programs" in *SAS/C Compiler and Library User's Guide, Fourth Edition*, for more information on **<resident.h>**.

312 **WARNING: Too many access method keywords specified.**
Errno Value: ELIMIT

Explanation

An access method parameter string specified more than the maximum number of keywords allowed. The maximum is currently 30.

313 **WARNING: Undefined file name style: [style]**
Errno value: EARG

Explanation

The file name style specified in the open is invalid. Valid styles are:

ddn: dd name
dsn: MVS dataset name
tso: TSO file name
cms: CMS file name
xed: XEDIT file
sf: CMS shared file
sfd: CMS shared file directory
path: NFS file (on another system)
hfs: OpenEdition HFS file

This message may be generated for an all-resident program if it tries to use a file name that is supported only if an appropriate symbol is defined when including **<resident.h>**. For instance, an all-resident program can access files with **sf** style file names only if the program first defines the symbol **RES_SHARED_FILE** before including **<resident.h>**.

Action

Refer to "I/O Functions" in Chapter 3 of the *SAS/C Library Reference, Volume 1*, for more details. See "All-Resident C Programs" in *SAS/C Compiler and Library User's Guide* for more information on **<resident.h>**.

314 **WARNING: Unknown or duplicate access method keyword: [string]**
Errno value: EARG

Explanation

An access method parameter string was specified with an unknown keyword or a repeated keyword. The start of the incorrect text is printed in the message.

Action

Refer to"I/O Functions" in Chapter 3 of the *SAS/C Library Reference, Volume 1*, for information on valid amparms.

315 **`WARNING: Improper file pointer, file not reopened.`**
`Errno value: EARG`

Explanation

The **`FILE`** pointer argument to **`freopen`** or **`afreopen`** is not a valid **`FILE`** pointer. This message may be generated if library work areas have been overlaid.

Action

Make sure a valid file pointer is passed.

---

316 **`WARNING: Attempt to use null FILE pointer.`**
`Errno value: ENOTOPEN`

Explanation

A null **`FILE`** pointer was passed to an I/O function. This is usually the result of failing to check whether a file was successfully opened before reading or writing it.

---

317 **`WARNING: File [name] was not opened successfully, error flag not cleared.`**
`Errno value: ENOTOPEN`

Explanation

`clearerr` or **`clrerr`** was called to clear the error flag for a file, but the flag could not be cleared. The file is a standard file (**`stdin, stdout`** or **`stderr`**) that failed to open successfully. A standard file can fail to open due to an invalid redirection, an invalid specification of **`_stdinm`**, **`_stdonm`**, **`_stdenm`**, or a missing DD statement.

Action

A previous message should have explained why the file failed to open. Take the action appropriate for this message.

---

318 **`WARNING: File [name] was not opened successfully.`**
`Errno value: ENOTOPEN`

Explanation

One of the standard files (**`stdin`**, **`stdout`**, or **`stderr`**) failed to open successfully. This can be caused by a missing DD statement, an invalid redirection or an invalid **`_stdinm`**, **`_stdonm`**, or **`_stdenm`** specification. Another message should have been generated describing the failure.

Action

A previous message should have explained why the file failed to open. Take the action appropriate for this message.

---

319 `WARNING: Error flag set permanently, not cleared.`
`Errno value: EPREV`

Explanation

An attempt was made to clear the file's error flag, but the error flag had been set permanently. This occurs when an error is so severe that no recovery is possible.

Action

The only means of clearing the error flag in this case is to close and reopen the file. A previous message should explain the cause of the original error. Note, by using the function **clrerr** to clear the error flag, you can determine at execution time whether or not error recovery is possible.

---

320 `WARNING: I/O not performed due to previous error.`
`Errno value: EPREV`

Explanation

A request was made to perform I/O on a file that was in an error state. The I/O operation was not performed.

Action

After detecting an error in file use, call the **clearerr** or **clrerr** function to reset the error flag. No use of the file (other than closing it) is allowed while the error flag is set.

---

321 `WARNING: Record contains more data than requested.`
`Errno Value: EFFORM`

Explanation

For a call to **afread**, the record to be read was larger than the amount specified by the **afread** arguments. For a call to **afwrite**, an existing record was being overwritten, and the amount to be written was less than the existing record size. (This can happen only when the file has **trunc=no** behavior.)

Action

Correct the program to specify a large enough number of items when calling **afread** or **afwrite**. See *SAS/C Library Reference* description of **afread** and **afwrite** for more details.

---

322 `WARNING: Function valid only for binary files.`
`Errno value: EUNSUPP`

Explanation

A call was made to **afread** or **afwrite** to access a file that was opened for text. These functions are only valid for files opened in binary mode.

Action

See the *SAS/C Library Reference* description of **afread** and **afwrite** for more details.

---

323 `WARNING: Record does not contain an integral number of items`
`Errno value: EFFORM`

Explanation

This message indicates that **fread** or **afread** was called to read items whose size was greater than 1, but the amount of data read was not an even number of items. For instance, reading 2 integers (size 4) from a file containing six bytes of data would cause this message to be generated.

Action

Correct the program, or correct the input data to conform to the program's requirements. Note that the second argument to **fread** or **afread** is the item size, and the third argument is the count. If you code **afread(buf, 6000, 1, fileptr)**, the call will fail unless the next input record is exactly 6000 bytes long. If you code **afread(buf, 1, 6000, fileptr)**, you can successfully read any record whose length is 6000 bytes or less.

---

324 **WARNING: Record too small to contain all items.**
Errno value: EFFORM

Explanation

A request was made using **afwrite** to output more data than could be contained in a single record.

Action

Increase the output file's record size, or correct the program to write smaller records. See *SAS/C Library Reference* description of **afwrite** for more details on the behavior of **afwrite**.

---

325 **WARNING: File not open for input.**
Errno value: EUNSUPP

Explanation

A request was made for input from a file that was not opened for input. The file's error flag is set.

---

326 `WARNING: File not open for output.`
`Errno value: EUNSUPP`

Explanation

A request was made for output to a file that was not opened for output. The file's error flag is set.

327 `WARNING: File does not support rewind.`
`Errno value: EUNSUPP`

Explanation

An attempt was made to rewind a file that does not support rewind. The file's error flag is set.

Action

See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

328 `WARNING: File does not support fseek.`
`Errno value: EUNSUPP`

Explanation

An attempt was made to seek on a file that does not support seeking. The file's error flag is set.

Action

See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

329 `WARNING: File does not support seek to EOF.`
`Errno value: EUNSUPP`

Explanation

A request was made to seek to end of file on a file (for instance, a PDS member) that does not support seeking to EOF. The file's error flag is set.

Action

See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

330 **`WARNING: No more characters can be pushed back.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to push back more characters with **`ungetc`** than can be supported by SAS/C. The ISO/ANSI standard states that only one character of pushback can be guaranteed. SAS/C allows more than one character to be pushed back when each character pushed back matches the character read from that position, except at the start of a record.

Action

Refer to the *SAS/C Library Reference* description of **`ungetc`** for more details.

---

331 **`WARNING: Value to be pushed back is not a valid character.`**
`Errno value: EARG`

Explanation

The argument to **`ungetc`** is neither a valid character or **`EOF`**. (That is, it is an integer outside the range of the **`char`** data type.)

Action

Correct the program to pass a **`char`** value or **`EOF`**.

---

333 **`WARNING: Read rejected, last file operation was a write.`**
`Errno value: EUSAGE`

Explanation

An input request was rejected because the last file operation was a write. The ISO/ANSI standard requires that a seek or flush operation be performed between write and read requests.

Action

Correct the program to call **`fseek(file, 0, SEEK_CUR)`** or **`(fflush(file)`** between the output operation and the input operation.

---

334 **`WARNING: Write rejected, last file operation was a read.`**
`Errno value: EUSAGE`

Explanation

An output request was rejected because the last file operation was a read. The ISO/ANSI standard requires that a seek operation be performed between read and write requests unless the read detected end of file.

Action

Correct the program to call **`fseek(file, 0, SEEK_CUR)`** between the output operation and the input operation.

---

335 `WARNING: Read rejected, file [name] is a write-only file.`
`Errno value: EUNSUPP`

Explanation

An input request was rejected because the file named in the message was opened as a write-only file.

Action

Either open the file for reading as well as writing, or correct the program not to read from the file.

---

336 `WARNING: Write rejected, file [name] is a read-only file.`
`Errno value: EUNSUPP`

Explanation

An output request was rejected because the file named in the message was opened as a read-only file.

Action

Either open the file for writing as well as reading, or correct the program not to write to the file.

---

337 `WARNING: Write rejected, file must be positioned to start before writing.`
`Errno value: EUSAGE`

Explanation

To switch between input and output modes, the program must be at the beginning of file for certain files, notably MVS PDS members. If this restriction is violated, the requested operation is not performed and the error flag is set for the file.

Action

See Table 3.6 in *SAS/C Library Reference, Volume 1* for information on file types for which seeking is restricted.

---

338 `NOTE: File to be opened does not support print format output, print=yes ignored.`
`Errno value: unchanged`

Explanation

The file to be opened does not have a RECFM (record format) that includes A (ANSI carriage control), and therefore the **`print=yes`** amparm cannot be supported. The open attempt will continue ignoring **`print=yes`**.

Note that the standard file **`stdout`** is opened by the library using the amparm **`print=yes`**. Therefore, this message may be issued for **`stdout`** if the file's RECFM does not include A, even though **`stdout`** is not explicitly opened by the program.

Action

None is required. However, to avoid the message either use a file capable of being printed or remove the **print=yes** amparm.

If the message is issued for **stdout**, you can use the external variable **_stdoamp** to override the library's amparms for this file. For example, you can code **char *_stdoamp = ''''** to specify that **stdout** should be opened without any amparms.

---

339 **WARNING: Write rejected, length of existing record cannot be changed.**
Errno value: EUSAGE

Explanation

An attempt was made to change a text file's record structure by changing a new line character to an ordinary character or changing an ordinary character to a new line. An example is trying to replace the text "abcn" with "ancn" or "abcd". The first attempts to replace one record with two, the second attempts to combine this record and the next record. The error flag is set for the file.

Note, this condition occurs only for **trunc=no** files, since with a **trunc=yes** file, the file is truncated at the point of output, and therefore the length of the last record can be changed.

Action

Correct the program either to open the file for binary access or to preserve the file's record structure. For CMS minidisk files that support both **trunc=yes** and **trunc=no**, consider using **trunc=yes**.

---

340 **WARNING: Improper value for access-method keyword: [keyword]=[value].**
Errno value: EARG

Explanation

An invalid value for the access-method keyword [keyword] was specified in an amparms string. Valid access-method keywords and their values are:

keyword	value
org	ps, os, po, pds, pdse, ks, es, rr, ls, byte
pad	no, null, blank
commit	no, yes
recfm	f, v, u
alcunit	trk, block, cyl, rec, krec, mrec
trunc	yes, no
grow	yes, no
rlse	yes, no
print	yes, no
order	seq, random
share	rls, rlsread, alloc, ispf

Valid access-method keywords with numeric (integer) values are:

reclen	**bufmax**	**keyoff**
blksize	**space**	**bufnd**
page	**extend**	**bufni**
bufsize	**dir**	**bufsp**
buflim	**keylen**	

The **reclen** amparm may also be specified as **reclen=x**.

Action

Correct the amparm. See *SAS/C Library Reference, Volume 1*, for more information on amparms.

---

341 **NOTE: The [keyword] keyword is ignored when opening a [type] file.**
Errno value: unchanged

Explanation

The amparm keyword cited cannot be honored for the type of file being opened, so the keyword is ignored. Some invalid keyword/file combinations are:

page and binary

pad and binary (if opened for input)

eof and binary terminal.

Action

Remove the amparm.

---

342 **WARNING: Internal error locating end of relative file.**
Errno value: ELIBERR

Explanation

The library was unable to correctly locate the end of file. This could be caused by invalid file sharing.

Action

Refer to IBM messages listed in the job log for other information, and contact SAS/C Technical Support.

---

343 **`WARNING: Record [num] of file does not have the correct length.`**
`Errno value: ESYS/ECORRUPT`

Explanation

The length of a record was not consistent with the record format of the file. This could be caused by using JCL DCB parameters to tell the library that a file was suitable for **`''rel''`** access when it fact it was not. The error flag is set for the file.

Action

The presence of a wrong-length record in a relative file may cause incorrect file positioning among other errors. Therefore, if you receive this message, you should be suspicious of the correctness of any I/O to the file after the incorrect record.

Refer to IBM messages listed in the job log for other information, and contact SAS/C Technical Support for further assistance if this condition is not associated with overriding the file's DCB parameters.

---

344 **`WARNING: Internal error: end of file occurred prematurely.`**
`Errno value: ELIBERR`

Explanation

End of file occurred unexpectedly while the library was processing a file opened for UNIX-style I/O. This could be caused by several users trying to access the same file simultaneously, which is generally not supported by MVS or CMS. In other circumstances, it is a SAS/C library internal error.

Action

Contact SAS/C Technical Support.

---

345 **`WARNING: File ''[name]'' not opened, read-only conflicts with truncate/exclusive.`**
`Errno value: EARG`

Explanation

When a file was opened using the **`open`** function for read-only access, one of the open options **`O_TRUNC`** or **`O_CREAT`**|**`O_EXCL`** was specified. These options are supported only for files that allow writing. The open failed.

Action

Correct the program to remove the unsupported options or to specify writing.

---

346 **`WARNING: File ``[name]'' not opened for exclusive access. File already exists.`**
`Errno value: EEXIST`

Explanation

The file to be opened already exists, but the **`open`** call required a non-existing file due to the use of **`O_CREAT`**|**`O_EXCL`**. The open failed.

Action

Assure that the file to be opened does not exist, or change the call to **`open`** not to specify **`O_EXCL`**.

---

347 **`WARNING: File ``[name]'' not opened, creation of temporary copy failed.`**
`Errno value: various`

Explanation

A file was opened that could not directly support UNIX-style I/O. For these files, the library creates a temporary file into which the file to be opened is copied. This temporary file failed to open. This failure is described by a previous message.

Action

Take whatever action is appropriate for the previous message. See Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume 1*, for more information on the SAS/C UNIX-style I/O implementation.

---

348 **`WARNING: [operation] rejected, file position unknown due to a previous error.`**
`Errno value: EPREV`

Explanation

The file position is unknown because of a previous error. This may be caused by an error occurring in a previous call to **`lseek`** or in any other operation that issued a seek.

Action

The problem can be corrected by calling **`lseek`** to set the file position to a previously determined location.

---

349 **`WARNING: Write suppressed due to previous read error.`**
`Errno value: EPREV`

Explanation

A write request using UNIX-style I/O failed due to a previous read error. Failing subsequent writes protects a partially unreadable file from further damage.

Action

Correct the previous read error, for which a previous message should have been issued.

---

350 **`WARNING: Open failed, too many open files.`**
`Errno Value: ELIMIT`

Explanation

The maximum number of files were opened simultaneously. The SAS/C library allows up to 256 files to be open at once. However, note that any call to **`open`** for a file that is not suitable for **`''rel''`** access creates two open **`FILE`** pointers.

Action

Correct the program to use fewer open files at one time.

---

351 **`WARNING: File to be closed is not a valid open file.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to close a **`FILE`** pointer that has already been closed, was never opened, or contains a corrupt **`FILE`** pointer value. If this message is received during program termination, it generally means that **`FILE`** pointers maintained by the library and used at program termination to close all open files have been corrupted. These **`FILE`** pointers are stored in the PRV (pseudo-register vector) for the main program load module. The most likely cause of such corruption is writing beyond the bounds of a **`__rent static`** or **`extern`** item.

Action

If the message was generated for a call to **`fclose`**, correct the argument to **`fclose`**. If the message is issued during program termination, check the program source for storing outside the bounds of **`extern`** or **`static`** data. Variables stored in the PRV immediately before the library pseudo-register **`L$CXGIOT`** are the most likely to be at fault. (A mapping of the PRV can be obtained from CLINK if the CLINK PREM option is used, and from the linkage editor if PREM is not used.)

---

352 `WARNING: No more temporary files can be opened.`
`Errno value: ELIMIT`

Explanation

A maximum of 255 temporary files can be opened simultaneously. This limit has been reached. Note that temporary files are opened by the library when UNIX-style I/O is requested for a file that does not support **''rel''** access, as well as when the **tmpfile** function is called.

Action

Change the program's logic so that fewer temporary files are open simultaneously.

---

353 `WARNING: Open failed, trunc=[value] not supported for this file.`
`Errno value: EFATTR`

Explanation

Certain file types do not support **trunc=yes**: VSAM, CICS Intrapartition transient data queues, CICS spool files, CMS shared files, and OpenEdition HFS files.

Certain file types do not support **trunc=no**: MVS PDS members and sequential data sets, including OS sim (filemode 4 and tape) files on CMS, except for those suitable for **''rel''** access opened as binary. However, if **grow=no** is used with these file types, only **trunc=no** is supported.

Action

Correct the program. For more information regarding the **trunc** amparm refer to the *SAS/C Library Reference*, Chapter 3, "I/O Functions."

---

354 `WARNING: I/O interrupted by signal, data may be lost.`
`Errno value: EINTR`

Explanation

A signal was received during an I/O operation, and a signal handler performed I/O to the same file or used **longjmp** to avoid returning to the point of interrupt. This leaves the file in an uncertain state, and an "interrupt flag" is set that prevents further I/O to the file until **clearerr** is called.

Action

Avoid use of files within signal handlers, and avoid the use of **longjmp** for signals that may occur while a library I/O function is running. If these cannot be avoided, add calls to **clearerr** as necessary to allow I/O to continue.

---

355 `WARNING: Invalid text in access method parameter string: [text]`
`Errno value: EARG`

Explanation

An item in the amparm string did not contain an equal sign. All amparms must have the form "keyword=value." Note that the syntax does not allow an amparm value to contain a comma.

---

356 `WARNING: Open failed, trunc=yes not supported by UNIX style I/O.`
`Errno value: EUSAGE`

Explanation

UNIX I/O is by nature not truncating. An attempt to use **trunc=yes** with binary UNIX-style I/O is not allowed.

Action

For more details on UNIX-style I/O and its implementation on MVS and CMS refer to *SAS/C Library Reference*, Chapter 3, "I/O Functions."

---

357 **WARNING: Open failed, grow=no is not compatible with the open mode.**
Errno value: EUSAGE

Explanation

In a call to **aopen**, the amparm **grow=no** was specified, but a conflicting open mode option was also set: one or more of **O_TRUNC, O_APPEND** or **O_CREAT**. The file was not opened.

Action

Remove the conflicting open mode settings, or remove **grow=no** from the amparms. For more details on the **grow=no** amparm, refer to *SAS/C Library Reference*, Chapter 3, "I/O Functions."

---

358 **WARNING: File not opened, the ''keylen'' amparm was specified for a text or binary stream.**
Errno value: EUSAGE

Explanation

The **keylen** access method parameter is valid only for VSAM files. If specified for a non-VSAM or LDS data set the **keylen** value must be 0.

Action

Correct the program or JCL so that a VSAM file is accessed, or modify the program not to specify a **keylen** other than 0. For more details on the **keylen** keyword refer to *SAS/C Library Reference*, Chapter 3, "I/O Functions."

---

359 **WARNING: This file is too large to process with the ''rel'' access method.**
Errno value: EFATTR

Explanation

A file opened for access using the **''rel''** access method is larger than $(2^{32} - 2)$ bytes. Because the file position is stored in a fullword and the bit pattern **0xffffffff** is reserved for an error indication, the file cannot be opened.

Action

Modify the application to allow the large file to be split into several files, or use the **afopen** function to force the use of the **''seq''** access method.

---

360 **WARNING: Output to file [name] lost due to previous error.**
Errno value: EPREV

Explanation

When a file opened for UNIX-style I/O was closed, the new file contents from the library's work file were not copied to the user's file. Due to this error, some data could not be copied.

Action

Refer to previous library messages and any IBM messages written to the job log for additional information on the failure, and take appropriate action.

---

361 `WARNING: I/O failure, the file position has overflowed.`
`Errno value: ELIMIT`

Explanation

An `lseek` or `fseek` caused the effective file position to become negative or greater than 2^{32}.

---

362 `WARNING: File does not support ftell except at start of record.`
`Errno value: EUNSUPP`

Explanation

For certain files accessed as binary (notably VSAM ESDS clusters), `ftell` is supported only when the file is positioned at the beginning of a record.

Action

Correct the program to use `ftell` only at record boundaries when processing a VSAM ESDS. See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

---

363 `WARNING: The file position cannot be accurately returned as a long integer.`
`Errno value: ERANGE`

Explanation

The library was unable to successfully convert the file position to a `long` integer, the return type of the `ftell` function. This indicates that the file being processed is too large or has records that are too large to allow all possible file positions to be returned successfully. See the "File Positioning" section in Chapter 3 of *SAS/C Library Reference, Volume 1*, for a detailed discussion of how file positions are transformed into `long` values.

Action

A possible alternative is to use `fgetpos/fsetpos` instead of `ftell/fseek` for files using standard I/O.

---

364 `WARNING: File does not support seeks relative to EOF.`
`Errno value: EUNSUPP`

Explanation

An attempt was made to seek on a file accessed using the **''seq''** access method with an offset from end of file other than 0. This is not supported.

Action

If it is necessary to seek relative to end of file, access the file using the **''rel''** access method, or use UNIX-style I/O. See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

---

365 **WARNING: File does not support seeks relative to current position.**
Errno value: EUNSUPP

Explanation

An attempt was made to seek on a file opened for text access using a non-zero offset relative to the current position. This is not supported.

Action

If seeking relative to the current position is required, consider whether the program can be modified to use binary access and the **''rel''** access method. See Table 3.6 in *SAS/C Library Reference, Volume 1*, for information on file types for which seeking is restricted.

---

366 **WARNING: Seek failed, current file position unknown due to previous error.**
Errno value: EPREV

Explanation

An attempt was made to seek relative to the current file position. Due to a previous error the file position is unknown and cannot be used to reposition the file.

Action

Correct the previous error or reposition the file to a previously determined location.

---

367 **WARNING: I/O request failed due to a previous seek error.**
Errno value: EPREV

Explanation

An I/O request was made that could not be completed because the file position was unknown due to a previous seek failure. A separate diagnostic should have been generated by the previous failure.

Action

Use the **fseek, fsetpos** or **rewind** function to define a valid file position after a seek failure.

---

368 **WARNING: File does not support ftell.**
Errno value: EUNSUPP

Explanation

The file or access method does not permit the use of **ftell**. For instance, **ftell** cannot be used with a file open for binary access unless its return value is the number of bytes from the start of file. Most files do not permit this value to be readily computed.

Action

Consider using **fgtepos** rather than **ftell**, which is supported for more kinds of file. See "File Positioning" in the *SAS/C Library Reference, Volume 1.*

---

369 **WARNING: File not opened, grow=no is valid only for a file opened with r+.**
Errno value: EUSAGE

Explanation

The amparm **grow=no** is incompatible with all open modes other than **''r+''**.

Action

Correct the program by removing **grow=no** from the amparms, or opening the file for **''r+''**. For more details on the **grow=no** amparm, refer to *SAS/C Library Reference*, Chapter 3, "I/O Functions."

---

370 **WARNING: File not opened. grow=no and trunc=yes are conflicting options.**
Errno value: EUSAGE

Explanation

The **grow=no** and **trunc=yes** access method parameters are incompatible. The file is not opened.

Action

Remove one of the two amparms.

---

371 **WARNING: Attempt to extend file opened with grow=no.**
Errno value: EUSAGE

Explanation

A file opened using the **grow=no** amparm cannot be extended. The existing data can be changed; however, adding new data is not permitted.

Action

Do not use the **grow=no** amparm if you intend to add new data to a file. For more details on the **grow=no** amparm refer to *SAS/C Library Reference* Chapter 3, "I/O Functions."

---

372 **WARNING: File not opened. commit=no is not supported for this file type.**
Errno value: EFATTR

Explanation

The **commit=no** amparm is supported only for files in the CMS shared file system, or files accessed via binary UNIX-style I/O.

Action

Use the **commit=yes** keyword only in situations where it is supported. For more details on the **commit=no** amparm, refer to *SAS/C Library Reference* Chapter 3, "I/O Functions."

---

373 **NOTE: The argument to [function] did not point to a valid open file.**

Errno value: unchanged

Explanation

The argument to the function indicated did not point to a valid open file. The file may have failed to open, or the argument may have been overlaid. This message is associated with file inquiry functions like **fattr** and **fnm**. After the message the function returns a degenerate value (e.g., the address of an all-zero **struct fattrib** object for **fattr**).

Action

Correct the program to pass a valid **FILE** pointer.

---

375 **WARNING: afflush rejected, last file operation was a read.**
Errno value: EUSAGE

Explanation

The **afflush** request was rejected because the last file operation was a read. Because **afflush** is considered an output operation, a seek must be performed between a read and a call to **afflush**.

Action

A call to **afflush** following a read request is highly unusual and may indicate a logic error. If the order of operations is correct, add a call to **fseek(file, 0, SEEK_CUR)** before the call to **afflush**.

---

376 **`WARNING: afflush failed, file not positioned to end of record.`**
`Errno value: EUSAGE`

Explanation

For some file types it is not possible to perform an **`afflush`** unless the file is positioned to the end of a record. For instance, this is the case for files opened with the amparm **`trunc=yes`**.

Action

Modify the program logic so that **`afflush`** is used only when positioned to the end of a record. For more information on afflush, see the function description in *SAS/C Library Reference, Volume 1.*

---

377 **`WARNING: Unable to write zero-length record to this file.`**
`Errno value: EUNSUPP`

Explanation

An attempt was made using the **`afwrite`** function to write a zero-length record to a file that does not support zero-length records. Nothing is written and the error flag is set for the file.

Action

Make sure the output file supports zero-length records (i.e., is an MVS RECFM VB file) before attempting to write one.

---

378 **`NOTE: Buffered input data discarded.`**
`Errno value: unchanged`

Explanation

The **`fflush`** function was called for an OpenEdition input file which does not support seeking, such as the terminal or a pipe. The call to **`fflush`** is successful, but one or more characters of input data were lost.

Action

Avoid the use of **`fflush`** on input files, or use the **`quiet`** function to suppress the diagnostic.

---

379 **`WARNING: I/O failure, record number too large.`**
`Errno value: ELIBERR`

Explanation

The **`''rel''`** access method detected an impossible record number during processing. This could occur processing a file larger than $2^{32} - 2$ bytes sequentially. Files this size are not supported by the **`''rel''`** access method.

Action

If the file is smaller than the supported maximum, contact SAS/C Technical Support. If the file is too large, process it using the **`''seq''`** access method. See *SAS/C Library Reference, Volume 1*, for more information on library access methods.

---

380 **`WARNING: Attempt to access non-standard stream with standard I/O function.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to access a non-standard stream, such as a stream opened using open mode **'k'**, with an unsupported standard I/O function.

Action

See the section "VSAM I/O" in Chapter 3 of the *SAS/C Library Reference, Volume 1*, for a list of the standard I/O functions that can be used with a keyed stream.

---

381 **WARNING: Attempt to use keyed access to a non-keyed stream.**
Errno value: EUSAGE

Explanation

An attempt was made to call a keyed I/O function such as **kretrv** for a text or binary stream.

---

382 **WARNING: This file does not permit searching.**
Errno value: EUNSUPP

Explanation

An attempt was made to call **ksearch** for a stream that did not permit searching. Searching is permitted only for keyed streams that allow reading. In particular, a stream opened for **''wk''** or **''ak''** does not permit searching.

---

383 **WARNING: Search key length is greater than file key length.**
Errno value: EARG

Explanation

The length of the key passed to **ksearch** is longer than the key length of the VSAM file. The request fails and the file's error flag is set.

Action

Correct the call to **ksearch**.

---

384 `WARNING: Replace/delete rejected, file opened for append.`
`Errno value: EUSAGE`

Explanation

A keyed file opened for append (open mode `''ak''` or `''a+k''`) does not support replacement or deletion of records. If this is required, the file should be opened with `''r+k''` or `''w+k''`.

Action

Correct the program.

---

385 `WARNING: Replace/delete rejected, current record not defined.`
`Errno value: EUSAGE`

Explanation

Before a VSAM record can be replaced or deleted, it must be retrieved using the `kretrv` function.

Action

Correct the program to retrieve a record before it attempts to delete or replace it.

---

386 `WARNING: Replace/delete rejected, current record not retrieved for update.`
`Errno value: EUSAGE`

Explanation

The most recently retrieved record from a VSAM file was retrieved with `K_noupdate` specified. The record must be retrieved for update before it can be replaced or deleted.

Action

Correct the program.

---

387 `WARNING: Record not replaced, the key has been changed.`
`Errno value: EUSAGE`

Explanation

It is not permitted to replace a record with another record that has a different key.

Action

Correct the program. The effect of replacing a record and changing its key can be achieved by adding the replacement record and then deleting the old record.

---

388 **`WARNING: Record not written, too large for file.`**
`Errno value: EUSAGE`

Explanation

A call to **`kreplace`** or **`kinsert`** specified a record that was larger than permitted by the file.

Action

Correct the program, or use a VSAM file with a larger maximum record length.

389 **`WARNING: Record not written, does not contain complete key.`**
`Errno value: EUSAGE`

Explanation

The record to be added or replaced was so small that some bytes of the key field were not present. The record was not written and the file's error flag is set.

390 **`WARNING: Record not deleted, specified key does not match.`**
`Errno value: EUSAGE`

Explanation

A call to the **`kdelete`** function was made that specified a key different from the key for the currently retrieved record. The record was not deleted, and the file's error flag was set.

Action

Correct the program. If this check is not wanted, modify the program to pass NULL as the key pointer to **`kdelete`**.

391 **`WARNING: This file does not permit records to be deleted.`**
`Errno value: EUNSUPP`

Explanation

The file did not allow deletion of records. In particular, it is not possible to delete records from an ESDS or from a path whose base cluster is an ESDS.

Action

If deletion of records is required, switch to some other kind of VSAM cluster.

392 **`WARNING: Position information unavailable for this file.`**
`Errno value: EUNSUPP`

Explanation

This condition should never occur.

Action

Report this message to SAS/C Technical Support.

393 **`WARNING: File position not meaningful, current record not defined.`**
`Errno value: EUNSUPP`

Explanation

A call to the **ktell** or **kgetpos** function requested position information about the current record. If there was no current record (i.e., the previous action on the file was not a call to **kretrv**), the call fails.

Action

Correct the program to call **ktell** or **kgetpos** only after a record has been retrieved.

---

394 `WARNING: This file does not support backwards inexact searches. Errno value: EUNSUPP`

Explanation

A **ksearch** call was made that specified **K_backwards** but not **K_exact**. These searches cannot be supported for files that allow duplicate keys.

Action

Modify the logic of the program, or use a file that does not permit duplicate keys.

---

395 `WARNING: This file does not support inexact searches. Errno value: EUNSUPP`

Explanation

A **ksearch** call was made that did not specify **K_exact**. The actual file is an ESDS that only supports exact searches.

Action

Correct the program, or use another type of cluster that supports inexact searches.

---

396 `WARNING: This file does not support generic key searches. Errno value: EUNSUPP`

Explanation

A **ksearch** call was issued that specified a generic key, but the file does not support generic searches. Generic searches are supported only for a VSAM KSDS.

Action

Correct the program, or use a KSDS rather than an ESDS or RRDS.

---

397 `WARNING: Searching not supported for output-only file.`
`Errno value: EUNSUPP`

Explanation

VSAM files support searching only when the open mode allows reading.

Action

Open the file for **``r+k''**, **``w+k''**, or **``a+k''** rather than **``wk''** or **``ak''**.

---

398 `WARNING: Library internal error, keyed file has zero length key.`
`Errno value: ELIBERR`

Explanation

This is either a library internal error or an indication that library control blocks have been corrupted.

Action

Contact SAS/C Technical Support.

---

400 `WARNING: File buffer not allocated.`
`Errno value: EUSAGE`

Explanation

This is a library internal error.

Action

Contact SAS/C Technical Support.

---

401 `WARNING: Operation not supported.`
`Errno value: EARG/EUNSUPP`

Explanation

An unsupported request has been made for I/O to a terminal input file. This is a library internal error.

Action

Contact SAS/C Technical Support.

---

402 `WARNING: Requested file number already associated with an open file: [filenum]`
`Errno value: EUSAGE`

Explanation

A request to open a file with SAS/C file number [filenum] failed because the file number was already assigned to another open file. This is an internal library error.

Action

Contact SAS/C Technical Support.

---

403 `WARNING: Buffer length too small for amput.`
`Errno value: EUSAGE`

Explanation

During file output, the number of characters written was greater than the size of the file's buffer. This is an internal library error.

Action

Contact SAS/C Technical Support.

---

404 `WARNING: Descriptor [filenum] is not associated with an open file or socket.`
`Errno value: EBADF`

Explanation

A call to a UNIX-style I/O function specified a file descriptor that was not associated with an open file or socket. This error may be the result of not checking a call to **`open`** or **`socket`** for failure. It may also be caused by overlaying a variable containing a file descriptor.

Action

Correct the program to use valid file and socket numbers.

---

405 `WARNING: File open for append not positioned to end of file before output.`
`Errno value: EUSAGE`

Explanation

A file opened for append was not positioned properly to end of file. This is an internal library error.

Action

Contact SAS/C Technical Support.

---

406 `WARNING: Old file name and new file name style prefixes are incompatible.`
`Errno value: EUSAGE`

Explanation

A call was made to the **`rename`** functions using different file name styles for the old and new names. This is not permitted.

Action

For more information on file style prefixes, refer to Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume 1.*

---

407 `WARNING: Attempt to update record not retrieved for update.`
`Errno value: EUSAGE`

Explanation

This is an internal library error in **kreplace**.

Action

Contact SAS/C Technical Support.

---

408 **WARNING: Seek with non-zero offset not supported for keyed file.**
Errno value: EUNSUPP

Explanation

This is an internal library error.

Action

Contact SAS/C Technical Support.

---

409 **WARNING: File position not defined, no record retrieved.**
Errno value: EUSAGE

Explanation

This is a library internal error in **ktell** or **kgetpos**.

Action

Contact SAS/C Technical Support.

---

410 **WARNING: Unable to retrieve record located by previous search.**
Errno value: ELIBERR

Explanation

An attempt was made to search for or get an input record from a keyed file, but the record, which had been located previously, can no longer be found. This is an internal library error.

Action

Contact SAS/C Technical Support.

---

411 `WARNING: Unsupported direction change for file with duplicate keys.`
`Errno value: EUNSUPP`

Explanation

While processing a file that allows duplicate keys, an attempt was made to change the direction of retrieval. That is, a call **kretrv** specifying **K_backwards** was made after a call that did not specify **K_backwards**, or vice versa. For files with duplicate keys, you can only change the direction using a successful call to **ksearch**.

Action

Correct the program, or run the program with files that do not allow duplicate keys. For more information on VSAM processing refer to Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume 1.*

---

412 `WARNING: Unable to retrieve record due to previous failed search.`
`Errno value: EPREV`

Explanation

Positioning in a VSAM file that permits duplicate keys is lost after any failure in a backwards search. A successful call to **ksearch** must occur before any more records can be retrieved.

Action

Perform a successful **ksearch** before attempting to retrieve any more records if a **ksearch** call specifying **K_backwards** fails.

---

430 `WARNING: File type does not permit [action].`
`Errno value: EUSAGE`

Explanation

An attempt was made to perform the indicated action on a file whose device type does not permit that action.

Action

Contact SAS/C Technical Support.

---

431 `WARNING: Second argument to access is not a valid access mode.`
`Errno value: EARG`

Explanation

The access mode that was passed as the second argument to the **access** function was invalid. Valid access modes are **R_OK** (check for read access), **W_OK** (check for write access), 0 (check for existence), and **R_OK** | **W_OK** (check for read and write access). For OpenEdition HFS files, **X_OK** (check for execute access) is also honored.

Action

Use a valid symbolic value as the second argument to **access**. See the **access** function description in *SAS/C Library Reference, Volume 1.*

---

460 `WARNING: [name] function not implemented by TCP/IP software.`
`Errno value: ENOSYS`

Explanation

A socket function was called that is not supported by the socket implementation. For instance, the **givesocket** function is not supported by OpenEdition integrated sockets.

Action

Do not use the unsupported function, or use the **setsockimp** function to select an implementation that supports the function. See *SAS/C Library Reference, Volume 2*, for more information on **setsockimp**.

---

461 **WARNING: Socket implementation has already been determined.**
Errno value: EUSAGE

Explanation

A call was made to the **setsockimp** function, but the socket implementation had already been determined. Only one call to **setsockimp** is permitted, and it must precede all other socket calls.

---

462 **WARNING: Unable to load specified socket implementation: [name]**
Errno value: EARG

Explanation

The library was unable to load the requested TCP/IP implementation load module. The failure should be described by a previous message. Perhaps the implementation name is misspelled, or perhaps the MVS library or CMS minidisk containing the load module is not accessible.

Action

Contact your local Systems Programming Staff to make sure you are following local procedures for access to the correct software.

---

470 **WARNING: Vendor-specific TCP/IP error condition ([name]).**
Errno value: ESYS

Explanation

An error condition named by [name] was set by the TCP/IP vendor's software for a function error return that is not recognized or mapped by the SAS/C TCP/IP library.

Action

Consult the TCP/IP vendor's documentation for an explanation of the error condition named.

---

471 **WARNING: TCP/IP software is down. Reason: [text].**
Errno value: ENETDOWN

Explanation

IBM TCP/IP software is not running. Either it has never been started, or has gone down and is currently unavailable.

Action

If IBM TCP/IP has been installed at your site, run the program again when TCP/IP has been started and available. This message can also be the the result of supplying (or defaulting to) the wrong TCP/IP address space name. The correct name can be supplied in a number of different ways. Refer to *SAS/C Library Reference, Volume 2*, for details.

Otherwise, insure that your site's TCP/IP vendor supplied LSCNCOM module is found in the SAS/C library transient search order before the IBM TCP/IP version of the LSCNCOM load module shipped with the SAS/C transient library.

---

472 `WARNING: Socket level not supported by TCP/IP software.`
`Errno value: ENOPROTOOPT`

Explanation

A `level` argument other than `SOL_SOCKET` was passed to the `getsockopt` or `setsockopt` function.

---

473 `WARNING: TCP/IP software caused unrecoverable socket library failure.`
`Errno value: ESYS`

Explanation

Further socket library processing was attempted after a previous TCP/IP software failure placed the socket library in an unrecoverable state.

Action

This is a secondary message. Action should be directed toward resolving problems associated with previous error messages. The socket library unrecoverable state can be reset by terminating and reinitializing the current SAS/C environment or by calling the `socktrm` function.

---

474 **`WARNING: Internal failure in socket library. Reason: [text].`**
`Errno value: ELIBERR`

Explanation

An internal error occurred in the SAS/C socket library or a critical internal function call failed. The usual reasons for these internal failures are errors returned by IUCV functions dealing with path establishment or message exchanges with the TCPIP address space.

Common failures include "iutpset rc=1" and/or "iutpconn return code = -1", which indicate a problem establishing IUCV communication with the TCP/IP address space. These conditions are known to occur in association with products that run multiple job steps in a single address space, such as some popular TSO multi-session products, if more than one of the job steps attempts to run a socket application.

Another failure that is occasionaly seen is "iutpset rc=4" or "iutpset rc=5", which indicates a memory protection or addressing execption has occurred in IBM TCP/IP IUCV Platform code attempting to establish the IUCV path.

Most other failures are rare and SAS/C Technical Support should be contacted for assistance.

---

476 **`NOTE: TCP/IP virtual machine is: [name].`**
`Errno value: unchanged`

Explanation

This is an informational message note generated in conjunction with other run-time TCP/IP error messages relating to IUCV communication failures with the named virtual machine or MVS address space name.

Action

Take action based upon other run-time error messages.

---

477 **`NOTE: CMSIUCV macro failed. RC=[num].`**
`Errno value: unchanged`

Explanation

A CMSIUCV macro returned return code **`num`**. This run-time message is most commonly seen when the CMSIUCV macro fails attempting the CONNECT operation to the TCP/IP virtual machine.

Action

Consult the appropriate manual for your system (*VM/ESA CMS Application Development Reference* or *VM/SP System Facilities for Programming*) for an explanation of CMSIUCV macro return code.

---

478 `NOTE: IUCV error. IPRCODE=[num].`
`Errno value: unchanged`

Explanation

An IUCV macro completed and stored a failing IPRCODE. Note, this run-time message is analogus to 477, but applies to MVS IUCV platforms. The message is most commonly seen when an IUCVMCOM (MVS IUCV Platform macro) CONNECT fails.

Action

Consult the appropriate IBM MVS TCP/IP programmers reference for an explanation of the IUCV return code. Alternatively, consult *VM/ESA CP Programming Services* or *VM/SP System Facilities for Programming*, since many of the IUCV IPR codes for MVS TCP/IP IUCV Platform Support are not currently documented in the *TCP/IP Programmer's Reference.*

---

479 `WARNING: Session with TCP/IP software timed out.`
`Errno value: ESYS`

Explanation

The TCP/IP address space/virtual machine has failed to return a response to the library IUCV message handler before it timed out.

Action

Determine the cause for the non-responsiveness of the TCP/IP address space or virtual machine.

---

480 `WARNING: Socket file descriptor [num] is not open.`
`Errno value: EBADF`

Explanation

The file descriptor that was passed to a socket function was not the number of a valid open socket. Perhaps the program failed to check the return value from the **`socket`** function.

---

481 `WARNING: [function] does not support socket operations.`
`Errno value: EUNSUPP`

Explanation

A UNIX-style I/O function, such as **`lseek`**, which is valid for file descriptors, but not for socket descriptors, has been called for a socket.

---

482 **`WARNING: I/O buffer [num] [count | length] is invalid.`**
`Errno value: EINVAL`

Explanation

The library detected an error in the buffer count for a TCP/IP vector I/O function call **`readv`** or **`writev`**, or the I/O buffer length is invalid (for a socket function that has a buffer length parameter).

Action

Determine the cause within the application and correct the invalid buffer count or length.

---

483 **`WARNING: Error condition set by TCP/IP software.`**
`Errno value: various`

Explanation

An error condition was detected and returned by the system TCP/IP software, causing an **`errno`** value to be stored.

Action

Examine the failing function and **`errno`** value set for more diagnostic information.

---

484 **`WARNING: File descriptor (number [num]) for socket rendered unusable by TCP/IP software failure.`**
`Errno value: ECONNABORTED`

Explanation

The TCP/IP software placed the socket descriptor in an ABORTED state.

---

485 **`WARNING: SAS/C sockets support only TCP/IP (AF_INET).`**
`Errno value: EAFNOSUPPORT`

Explanation

An addressing family other than **`AF_INET`** was specified. Only **`AF_INET`** is supported for non-integrated sockets.

Action

Change the program to specify **`AF_INET`**. If you are using integrated sockets, **`AF_UNIX`** may also be specified.

---

486 **`WARNING: [num] is an incorrect socket address length for AF_INET.`**
`Errno value: EINVAL`

Explanation

The length of the socket address passed to **`bind`**, **`connect`**, or as part of the message header for **`sendmsg`** does not match the required length of **`sizeof(struct sockaddr_in)`**.

---

487 `WARNING: F_SETFD must be 0 or 1.`
`Errno value: EINVAL`

Explanation

The third argument for the `fcntl` function must be 0 or 1 when the second argument is `F_SETFD` and the first argument is a socket descriptor.

488 `WARNING: fcntl command unrecognized or not valid for socket.`
`Errno value: EINVAL`

Explanation

A call to `fcntl` for a socket specified an unsupported command or option. Only the commands `F_GETFD, F_SETFD, F_GETFL` and `F_SETFL` are supported, unless OpenEdition integrated sockets are used.

Action

See *SAS/C Library Reference, Volume 1*, for more information on the `fcntl` function.

489 `WARNING: [parmname] parameter cannot be NULL.`
`Errno value: EINVAL`

Explanation

An invalid NULL pointer was passed to the indicated function.

490 `WARNING: [parmname] parameter cannot be negative.`
`Errno value: EINVAL`

Explanation

An invalid negative parameter value was passed to the indicated function.

491 `WARNING: [parmname] parameter invalid.`
`Errno value: EINVAL`

Explanation

An invalid parameter value was passed to the indicated function.

492 `WARNING: [parmname] parameter not consistent with option or command.`

`Errno value: EINVAL`

Explanation

The indicated parameter value was inconsistent with the other function parameters.

494 **`WARNING: Library socket limit exceeded.`**
`Errno value: EMFILE`

Explanation

The library was unable to open any more sockets. If OpenEdition is installed, the maximum number of sockets is controlled by OpenEdition, and can be changed by modifying the OpenEdition startup parameters in SYS1.PARMLIB. The limit is 256 if OpenEdition is not installed.

Action

Modify the program to keep fewer sockets open at one time. If your site uses OpenEdition, ask your Systems Programming Staff to increase the OpenEdition socket limit.

495 **`NOTE: At most MAXNS (6) nameservers may be specified.`**
`Errno value: 0`

Explanation

More than MAXNS (6) NSINTERADDR nameserver definition keywords appear in the IBM TCPIP.DATA file. Only the first MAXNS nameservers found will be used by the resolver functions defined in **`<resolv.h>`** header file.

Action

Remove any unneeded NSINTERADDR statements from the TCPIP.DATA file. Having more than two or three name servers to query can result in poor performance for the TCP/IP resolver functions unless the servers are properly configured.

If the TCPIP.DATA file is shared among different systems it may be possible to place a system id prefix to identify the system(s) that use a particular name server and thus reduce the name server count on a per system id basis.

Alternately, the name server configuration may be changed so that the excess name servers are pointed to by other name servers in the in the configuration, thus reducing the number of NSINTERADDR statements required.

---

496 **`NOTE: Error in TCPIP.DATA file before line [num], columm [num]`**
`Errno value: 0`

Explanation

A syntax error or unrecognized keyword was encountered while processing the IBM TCPIP.DATA file. The line number and column of the statement are part of the message. The failing statement is ignored and processing of the TCPIP.DATA file continues with the next valid TCPIP.DATA statment

Action

Correct the TCPIP.DATA statement syntax problem identified by the message. If the syntax is correct, and the keyword is a valid TCPIP.DATA keyword, contact SAS/C Technical Support for assistance.

---

497 `NOTE: Could not determine system name for TCPIP.DATA .`
`Errno value: 0`

Explanation

The library was unable to identify the specific system executing the application.

For MVS this means the sytem name was not obtainable from the VMCF entry in the subsystem name table defined by IEFSSNxx member(s) of SYS1.PARMLIB.

For CMS, the SYSTEM NETID file is not accessable and/or failed to contain an entry name name for the current CPUID, and the QUERY USERID command failed to return a recognizable system name for the current userid.

Action

Consult the appropriate IBM TCP/IP Customization manual for your system for details on the system name specification for TCPIP.DATA files.

---

498 `WARNING: Internal error: attempt to release unused socket number [num]`
`Errno value: ELIBERR`

Explanation

This is an internal error in socket number management.

Action

Report the problem to SAS/C Technical Support.

---

499 `WARNING: TCP/IP software limit reached on number of sockets.`
`Errno value: EMFILE`

Explanation

The application is already using the maximum number of socket descriptors permitted by your system. No more socket descriptorsmay be allocated until some are closed.

Action

Close socket descriptors which are no longer in use before trying to obtain another.

---

500 `WARNING: File not [text], ddname [name] not defined.`
`Errno value: ENFOUND`

Explanation

The DDname specified was not defined and so could not be opened, removed, or renamed.

Action

Define the DDname via JCL, a TSO ALLOCATE command or a CMS FILEDEF.

---

500 **`WARNING: File not [text], transient data queue name [name] not defined.`**
`Errno value: ENFOUND`

Explanation

The transient data queue [name] was not defined to CICS and so could not be opened.

Action

Define the queue name in the CICS Destination Control Table (DCT).

---

501 **`WARNING: File not opened, terminal not defined in batch.`**
`Errno value: ENFOUND`

Explanation

An attempt was made to open the terminal (filename **`''*''`**) in MVS batch. SAS/C does not support a terminal or pseudo-terminal input file in batch.

Action

Do not attempt terminal input in batch. The **`intractv`** function can be used to tell whether the program is running in batch, as described in the *SAS/C Library Reference, Volume 2.*

---

501 **`WARNING: Internal error in l$cgbcm: Library filetype: [text]`**
`Errno value: ELIBERR`

Explanation

The library was attempting to locate a member within a MACLIB or TXTLIB. The library found GLOBALed MACLIBs or TXTLIBs but could not search them.

Action

Contact SAS/C Technical Support.

---

501 **`WARNING: Cannot specify file open mode ''[spec]'' for an Extrapartition destination.`**
`Errno value: EUNSUPP`

Explanation

The file open mode [spec] cannot be specified for this type of file. Extrapartition data queues and JES spool files can be opened for input or output but not both.

Action

Change the file mode argument string to indicate whether the file is to be read or written.

---

502 `WARNING: System macro [name] failed with return code [num] Errno value: ESYS/ELIBERR`

Explanation

A system macro or servce failed in an unexpected or unusual way. and the library did not recognize this specific error. The return code from the failing service is shown in the message.

Action

To find a more detailed explanation of the reason for the failure refer to the related documentation.

The following is a list of IBM publications that describe the return codes for particular macros and services used by the library.

Macro or Service	IBM Reference Documentation
ATTACH IDENTIFY	*MVS/ESA Application & Development Guide*
SWAREQ	*MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs*
BLDL ISITMGD OPEN STOW TCLOSE (see CLOSE TYPE=T)	*MVS/DFP Macro Instructions for Data Sets*
CATALOG (see CAMLST CAT) LOCATE OBTAIN RDJFCB RENAME SCRATCH UNCATALOG (see CAMLST UNCAT)	*MVS/DFP System Programming Reference*
IKJCT441 IKJSCAN PUTGET PUTLINE STACK TGET TPUT	*TSO/E Programming Services*

Macro or Service	IBM Reference Documentation
IRXINIT IRXSAY IRXSTK IRXSUBCM	*TSO/E REXX*
GLOBALV	*CMS Command Reference*
FSSTATE	*CMS Application Development Reference for Assembler*
BPXWRBLD	*Using REXX to Access OpenEdition MVS Services*

If the problem cannot be determined by referring to the related documentation, contact SAS/C Technical Support.

502 **`WARNING: EXEC CICS command ''[name]'' failed with response code [num].`**
`Errno value: ESYS`

Explanation

The run-time library issued an EXEC CICS command which received an unexpected exceptional response return code from CICS.

Action

The command and response code will be a guide for the action you should take. The response code can be looked up in the appropriate CICS documentation under EIBRESP. They are also listed after the command descriptions. Common responses include NOTAUTH, QIDERR, SYSIDERR, ISCINVREQ, IOERR, and LENGERR. If necessary, contact your CICS Systems Administrator or call SAS/C Technical Support.

503 **`WARNING: File never created, open failed.`**
`Errno value: ENFOUND`

Explanation

An attempt was made to open an MVS sequential file with open mode **`''r''`**, **`''rb''`**, **`''r+''`**, or **`''r+b''`**. The file contained no data and appeared never to have been initialized. SAS/C treats such files as "not existing."

For a file to be "created," it must not only be defined by the operating system but must also contain data (as recorded in the VTOC) or, for a VSAM file, have been initialized by writing at least one record to the file.

This message can be generated when a data set was created by the operating system via a DD card or TSO ALLOCATE command immediately before the program containing the failing open was run.

Action

Refer to Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume I*, for a detailed discussion of the SAS/C interpretation of "file existence." To correct the problem, use a file type for which an empty file can be distinguished from an uninitialized file, such as a PDS member, or ensure that the input file contains at least a byte of data before attempting to read it.

503 `WARNING: File ''[fname ftype fmode]'' not found.`
`Errno value: ENFOUND`

Explanation

An attempt was made to open a file in `''r''` or `''r+''` mode (or some other open mode beginning with `'r'`), and the file was not found. Either the file does not exist or the file has been FILEDEF'd but no data has been written to the file. CMS does not support the existence of files containing no characters.

Action

Make sure the file name is correct and, if the file is being opened for `''r''` or `''r+''` access, that the file has been created and contains data. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*.

---

504 `WARNING: PDS member not found: [member-name]`
`Errno value: ENFOUND`

Explanation

An attempt was made to open a PDS member for input using one of the open modes `''r''`, `''r+''`, `''rb''`, or `''r+b''`, but the member does not exist.

Action

Specify an existing PDS member, or change the open mode to one which allows a new member to be created.

---

504 `WARNING: Cannot open an [type] only Extrapartition destination for [operation].`
`Errno value: EUNSUPP`

Explanation

Either you are trying to read an extrapartition transient data destination defined to CICS as TYPEFLE=OUTPUT or you are trying to write to an extrapartition transient data destination defined to CICS as TYPEFLE=INPUT.

Action

Make sure the arguments to open match the transient data queue definitions in the CICS Destination Control Table (DCT).

---

505 `WARNING: File contains incorrect V-format control data.`
`Errno value: ECORRUPT`

Explanation

Processing of a RECFM V file has failed due to incorrect record or block length information in the file. This error indicates that either the DCB of the file is incorrect, or the file's contents have been corrupted.

Action

Investigate whether the file has been corrupted and should be recreated. This error can occur at the end of a file when the program that wrote the file ran out of disk space.

---

506 `WARNING: Permanent file error, reopen of ddname [name] failed.`
`Errno value: ESYS`

Explanation

During a call to **`rewind`** or a similar function, the library closed and reopened the file to reposition it. The file failed to reopen.

Action

Specifying FREE=CLOSE in the JCL can cause this error. If FREE=CLOSE was not specified, contact SAS/C Technical Support.

---

506 `WARNING: File not opened, transient data destination [name] is not open.`
`Errno value: EDEVICE`

Explanation

CICS returned an indication that the extrapartition queue [name] is not open.

Action

Modify the CICS Destination Control Table (DCT) definition for the queue so that it is opened when CICS initializes, or issue a CEMT SET TDQUEUE OPEN command for the named destination.

---

507 **`WARNING: A PDS member cannot be opened for append.`**
`Errno value: EFATTR`

Explanation

A PDS/PDSE member cannot be opened for append (open mode **`''a''`**, **`''ab''`**, **`''a+''`**, or **`''a+b''`**). This is an MVS restriction associated with the organization of PDS's.

Action

If this ability is required, use UNIX-style I/O to process the member. For more information, refer to Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume I*.

---

507 **`WARNING: A PDS member cannot be opened for output.`**
`Errno value: EFATTR`

Explanation

Under CMS a MACLIB or TXTLIB member can only be opened for input.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I* for information on filename specification under CMS.

---

507 **`WARNING: File not opened, reclen=x not supported for [file type].`**
`Errno value: EFATTR`

Explanation

The program specified **`reclen=x`** as an amparm to **`afopen`**. This specification is not supported for transient data queues or JES spool files.

Action

Either remove the **`reclen=x`** amparm or change it to a supported record length.

---

508 **`WARNING: File not opened, ''[name]'' is not a valid ''[style]'' style file name.`**
`Errno value: EARG`

Explanation

The file name shown does not have the correct syntax for a "[style]" style file name. (For instance, a **`''ddn''`** style name contains a period.)

A common cause of this error is a program ported from another system to MVS, where the default file name style is **`''ddn''`**, since most "portable" file names are not valid DDnames.

Action

Correct the file name. If interpreting file names as DDnames is inappropriate for your program, define the external variable **`_style`**, as described in the *SAS/C Compiler and Library User's Guide.*

If your program is a CICS application using a **`''td''`** or **`''spl''`** style file name, see the *CICS User's Guide* for more information.

---

509 `WARNING: File not opened, not found on specified volume.`
`Errno value: ECORRUPT`

Explanation

The file to be opened was not found on the correct volume. The volume may have been specified in the open amparms or the JCL, or it may have been obtained from the system catalog.

Action

If the program or JCL is incorrect, correct it. If the system catalog addresses an incorrect volume, contact your site's Systems Programming Staff for assistance.

---

509 `WARNING: File ''[fname ftype fmode]'' has invalid directory data.`
`Errno value: EFATTR`

Explanation

The library attempted to scan a MACLIB or TXTLIB to determine if the member to be opened existed. Invalid data was found within the library header, and the open failed.

Action

Contact your site's Systems Programming Staff for assistance in determining why the library is corrupt.

---

509 `WARNING: File not opened, remote transient data destinations not supported.`
`Errno value: EUNSUPP`

Explanation

Remote transient data destinations are not supported.

Action

Run the application on the remote system so that the named destination is logically a local one, or define the remote destination on the local system.

---

510 **`WARNING: Member ``[name]'' not found in globaled [type].`**
`Errno value: ENFOUND`

Explanation

An attempt was made to open a **`%MACLIB`** or **`%TXTLIB`** file name, but the search of the GLOBALed MACLIBs or TXTLIBs for the specified member failed to find it.

Action

Make sure the MACLIBs or TXTLIBs in the GLOBAL list contain the specified member and make sure the member name is spelled correctly.
Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for filename specification under CMS for use of **`%MACLIB`** and **`%TXTLIB`** file names.

---

510 **`NOTE: blksize=[num] keyword ignored, actual block size of file is [num].`**
`Errno value: unchanged`

Explanation

The program specified a **`blksize`** amparm, and the value specified was greater than the actual blocksize of the file. The specified blocksize is ignored.

Action

Reconcile the **`blksize`** amparms value with the blocksize of the actual file.

---

511 **`NOTE: Requested record length of [num] too large, reduced to [max].`**
`Errno value: unchanged`

Explanation

The maximum record length for an MVS data set is 32756. The maximum record length for a CICS spool file is 32760. A larger **`reclen`** was specified, but this was treated by the library as the maximum permitted.

Action

Correct the program to use a smaller record size, or use **`reclen=x`**.

---

512 **`NOTE: Memory not available for another [num]K DIV window. Performance may be affected.`**
`Errno value: unchanged`

Explanation

The library failed to allocate [num]K bytes of additional memory, which were needed to read or write a VSAM DIV window. The requested I/O will still be performed; however, the library will be forced to remap a previously allocated window.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for further information on VSAM linear data sets (DIV objects).

---

512 **`WARNING: File [fname ftype fmode] cannot be positioned to record number [num]`**
`Errno value: EUSAGE`

Explanation

A request was made to reposition the specified file. When the library attempted to position the file it determined that the position requested was not valid. This may be caused by unintentionally writing over the argument to **`fseek`** or **`fsetpos`**.

Action

Make sure the file position is valid. If an **`fseek`** or **`fsetpos`** argument has become corrupted, consider the use of the SAS/C debugger **`monitor`** command to locate the problem. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for details on file positioning in CMS.

---

512 **`WARNING: File not opened, unknown transient data destination type.`**
`Errno value: ESYS`

Explanation

The transient data destination associated with this file is not extrapartition, intrapartition, or remote.

Action

The run-time library only recognizes one of the above types of transient data destinations. One possibility is that the internal CICS transient data control information has been overwritten with misleading information. If that is not the case ensure that the destination definition in the CICS Destination Control Table (DCT) is correct.

---

513 **`NOTE: Memory not available for program specification bufsize=[num]. Actual bufsize is [num]K.`**
`Errno value: unchanged`

Explanation

The library was unable to allocate a buffer for a DIV data set of the size specified by the **`bufsize`** amparm. A smaller window size was used as indicated in the message.

Action

Specify another **`bufsize`** value or run the program in a larger region.

---

513 **`WARNING: Requested record length of [num] too large, may not exceed 65535.`**
`Errno value: EFATTR`

Explanation

The program specified a **`reclen`** amparm greater than the maximum of 65535. This is a restriction imposed by CMS.

---

513 `WARNING: I/O not performed, transient data destination ''[name]'' is not open.`
`Errno value: ENOTOPEN`

Explanation

The transient data destination is not in CICS OPEN status. An EXEC CICS command returned a response code of NOTOPEN.

Action

Modify the CICS Destination Control Table (DCT) definition for the queue so that it is opened when CICS initializes or issue a CEMT SET TDQUEUE OPEN command for the named destination.

---

514 `WARNING: File not opened, reclen for PRINTER may not exceed [num]`
`Errno value: EFATTR`

Explanation

The maximum record length for a PRINTER file is 230.

Action

Correct the program. If the application requires a **`reclen`** value greater than the maximum, do not specify a PRINTER file.

---

514 `WARNING: File not opened, no transient data destination assigned to stdin.`
`Errno value: ENFOUND`

Explanation

The program attempted to read from **`stdin`**. In CICS, unlike other environments, there is no defined standard input file. One must be explicitly specified by the program.

Action

Explicitly open **`stdin`** and specify a filename with the appropriate prefix or initialize the **`_stdinm`** variable with a default prefix and file name.

---

515 `WARNING: Internal error in [name]: Memory overlay on FCB.`
`Errno value: ELIBERR`

Explanation

While processing a file the library detected that the file was not opened correctly. The most likely cause of this is a memory overlay on the FCB.

Action

Contact SAS/C Technical Support.

---

515 **`WARNING: EXEC CICS [text] command failed with response code [num] and resp2 code [num].`**
`Errno value: ESYS`

Explanation

The run-time library issued an EXEC CICS command that received an unexpected exceptional response return code from CICS.

Action

The command and response codes will be a guide for the action you should take. The response code can be looked up in the appropriate CICS documentation under EIBRESP. They are also listed after the command descriptions. Common responses include NOTAUTH, QIDERR, SYSIDERR, ISCINVREQ, IOERR, and LENGERR. Discuss further with your CICS Systems Administrator or call SAS/C Technical Support.

---

516 **`WARNING: File not opened, disk [letter] is not write-accessed.`**
`Errno value: EFATTR`

Explanation

A request was made to open a file for output to a disk that is not write accessed.

Action

Correct the program or make the specified disk write accessible.

---

517 **`WARNING: File cannot be accessed using the [name] access method.`**
`Errno value: EFATTR`

Explanation

An explicit access method was requested for a file or device which cannot support it.

Action

This is most often caused by attempting to use the **`''rel''`** access method on a file that does not have RECFM F or RECFM FBS. The access methods and their requirements are:

term	for the terminal (or the SYSTERM DD statement in MVS batch)
rel	for binary access to RECFM F and RECFM FBS sequential files. **`''rel''`** is also accepted for RRDS and LDS VSAM files, and for OpenEdition disk files.
kvs	for VSAM files other than linear accessed using a keyed stream
fd	for OpenEdition files
seq	for anything remaining. **`''seq''`** is accepted for a terminal file, and **`''term''`** is substituted.

For more information on access methods refer to "I/O Functions" in *SAS/C Library Reference, Volume I*.

---

518 `WARNING: File not opened, member [name] not found in file ''[fname ftype fmode]''`
`Errno value: ENFOUND`

Explanation

An attempt was made to open a member of a MACLIB or TXTLIB for input; however, the member was not found in the specified library.

Action

Make sure that the member name is spelled correctly and that the library specified contains the member.

---

519 `WARNING: Invalid file identifier: ''[fname ftype fmode]''`
`Errno value: EARG`

Explanation

The specified file identifier is invalid. The filename, filetype, or filemode does not conform to CMS standards.

Action

Correct the file identifier. Refer to the *CMS User's Guide* for an explanation of the rules for valid CMS fileids.

---

520 `NOTE: Read terminated by attention.`
`Errno value: unchanged`

Explanation

The library was performing a terminal read operation when it received an attention. The read was terminated with possible loss of data.

Action

None required. The message is simply a reminder that any data entered in the transmission that ended with an attention may have been lost.

---

521 `WARNING: File ''[fname ftype fmode]'' is not a library.`
`Errno value: EFATTR`

Explanation

While attempting to process the GLOBALed list of MACLIBs or TXTLIBs, the open routine found a file that is not a library. This file is ignored.

Action

Correct the GLOBALed list of MACLIBs or TXTLIBs.

---

522 **WARNING: File not opened, the specifications ''recfm=f'' and ''reclen=x'' are incompatible.**
Errno value: EARG/EUSAGE

Explanation

EARG The amparms specified both **reclen=x** and **recfm=f**. **reclen=x** requires **recfm=v**.

EUSAGE The amparms specified **reclen=x**, but the actual RECFM of the data set was F.

Action

Do not specify **reclen=x** for a file unless the RECFM is V.

---

523 **NOTE: Program specifications ''reclen=[num]'' ignored for existing file. Actual LRECL is [num]**
Errno value: unchanged

Explanation

The program specified the amparm **reclen** with a record length different from the actual record length of the file being opened. The **reclen** of a file can only be changed when the file is opened for output (open mode **''w''**, **''wb''**, etc.).

---

524 **WARNING: File not opened, ''reclen=[spec]'' is inconsistent with other file attributes.**
Errno value: EFATTR

Explanation

The **reclen** amparm specified a record length that is inconsistent with other characteristics of the file being opened. For instance, the file has RECFM F, but the **reclen** does not evenly divide the file's block size.

Action

Correct the program, or use a file with the required record length.

---

524 **NOTE: Program specifications ''recfm=[spec]'' ignored for existing file. Actual RECFM is [spec]**
Errno value: unchanged

Explanation

The program specified the amparm **recfm** with a record format different from the actual RECFM of the file being opened. The RECFM of a file can only be changed when the file is opened for output (open mode of **''w''**, **''wb''**, etc.).

---

525 **`WARNING: File not opened, program specification ''reclen=x'' is incompatible with FILEDEF option RECFM F.`**
`Errno value: EFATTR`

Explanation

The program specified the amparm **`''reclen=x''`**, but the CMS FILEDEF statement specified a RECFM of F. A file with a RECFM of F must have a specified logical record length of 32760 or less.

Action

Correct the program or modify the FILEDEF.

---

526 **`WARNING: File not opened, [type] I/O not support by device.`**
`Errno value: EFATTR`

Explanation

The [type] can be either "update" or "append." If the type is "update" then the file being opened was a unit record device. These types of devices do not support update I/O. If the type is "append" the file being opened is probably a CMS tape file. SAS/C does not support appending to a tape file on CMS.

Action

Correct the program, or correct JCL or FILEDEFs not to use a device that is incompatible with the program.

---

527 **`WARNING: PDS member already exists, cannot be written with DISP=MOD.`**

`Errno value: EFATTR`

Explanation

An attempt was made to open an existing PDS member for which DISP=MOD was specified. When used with a PDS member, the JCL specification DISP=MOD is interpreted by MVS as requesting a member which does not yet exist.

Action

Either use DISP=OLD/SHR or delete the member first.

---

528 **`NOTE: FILEDEF option RECFM [spec] ignored for existing file. Actual RECFM is [spec]`**
`Errno value: unchanged`

Explanation

A CMS FILEDEF specified a RECFM which is different than the file's actual RECFM. The library, while opening the file, noticed the inconsistency and used the actual RECFM.

Action

None required. To suppress the message, correct the FILEDEF or call the **`quiet`** function before calling **`fopen`**. See the *SAS/C Library Reference, Volume 1*, for information on **`quiet`**.

---

529 **`NOTE: FILEDEF option LRECL [num] ignored for existing file. Actual LRECL is [num]`**
`Errno value: unchanged`

Explanation

A CMS FILEDEF specified a LRECL for a file being opened which was different from the file's actual LRECL. The library, while opening the file, noticed the inconsistency and used the actual LRECL.

Action

None required. To suppress the message, correct the FILEDEF or call the **`quiet`** function before calling **`fopen`**. See the *SAS/C Library Reference, Volume 1*, for information on **`quiet`**.

---

530 **`WARNING: File not opened, FILEDEF command failed with return code [num]`**
`Errno value: ESYS`

Explanation

The program attempted to open an OS simulated CMS file (filemode 4). The library attempted to create a FILEDEF for the OS file and the FILEDEF command failed.

Action

Look up the FILEDEF return code in the *CMS Command Reference* and take appropriate action..

---

531 **`NOTE: Program specification ``recfm=[spec]'' ignored for device. Actual RECFM is [spec]`**
`Errno value: unchanged`

Explanation

The program specified the amparm **`recfm`** with a record format different from the record format supported by the device being opened. The actual device format will be used.

Action

None required. To suppress the message, correct the amparms or call the **`quiet`** function before calling **`fopen`**. See the *SAS/C Library Reference, Volume 1*, for information on **`quiet`**.

---

532 `NOTE: Program specification ''reclen=[num]'' ignored for device. Actual LRECL is [num]`
`Errno value: unchanged`

Explanation

The program specified the amparm **reclen** with a record length different from the record size supported by the device being opened. The device's record size will be used.

Action

None required. To suppress the message, correct the amparms or call the **quiet** function before calling **fopen**. See the *SAS/C Library Reference, Volume 1*, for information on **quiet**.

---

533 `NOTE: FILEDEF option RECFM [spec] ignored for device. Actual RECFM is [spec]`
`Errno value: unchanged`

Explanation

A CMS FILEDEF specified a RECFM which differed from the record format supported by the device being opened. The library, while opening the device, noticed the inconsistency and used the device format.

Action

None required. To suppress the message, correct the FILEDEF or call the **quiet** function before calling **fopen**. See the *SAS/C Library Reference, Volume 1*, for information on **quiet**.

---

534 `NOTE: FILEDEF option LRECL [num] ignored for device. Actual LRECL is [num]`
`Errno value: unchanged`

Explanation

A CMS FILEDEF specified a LRECL, which is different from the record length supported by the device being opened. The library, while opening the device, noticed the inconsistency and used the record length supported by the device.

Action

None required. To suppress the message, correct the FILEDEF or call the **quiet** function before calling **fopen**. See the *SAS/C Library Reference, Volume 1*, for information on **quiet**.

---

535 **`NOTE: Blksize value [num] too large for device, reduced to [max]`**
`Errno value: unchanged`

Explanation

The library determined that the block size requested by the **`blksize`** amparm for a file being opened was larger than the device could support. The blocksize was reduced to the maximum [max] for the device.

Action

Specify a smaller block size, or avoid running the program using devices that cannot accomodate the block size required.

---

536 **`NOTE: Program specification ''[keyword]=[value]'' ignored for existing file. Actual [keyword] is [value].`**
`Errno value: unchanged`

Explanation

The library detected an incompatibility between the file characteristics requested via amparms and the actual characteristics of the file being opened. The program's request was ignored, and the existing data set's characteristics were retained.

Action

None required. Refer to Chapter 3, "I/O Functions" in *SAS/C Library Reference, Volume I* for information on use of the **`quiet`** function to suppress the message. See the section "Opening Files" for a detailed discussion of when the library's processing of amparms for existing files.

---

537 **`WARNING: File not opened, ''recfm=[spec]'' is inconsistent with other file attributes.`**
`Errno value: EFATTR`

Explanation

The record format specified by the program's amparms was not compatible with the other attributes of the file to be opened. For instance, the program specified **`recfm=f`**, but the file's LRECL did not evenly divide the BLKSIZE.

Action

Correct the program, or use a file which is compatible with the program's amparms.

---

538 **`WARNING: DISP=MOD not supported for multi-volume output files.`**
`Errno value: EFATTR`

Explanation

The SAS/C library does not support use of multi-volume output files with DISP=MOD.

Action

Consider the use of append mode (open mode **`''a''`** or **`''ab''`**) rather than DISP=MOD for these files.

---

538 **`WARNING: Unit exception bit set while accessing [name]`**
`Errno value: EDEVICE`

Explanation

An I/O operation to the named file resulted in a hardware device error.

Action

Contact your site's Systems Programming Staff.

---

538 `WARNING: Physical record size of [num] split to accomodate requested record size of [num].`
`Errno value: ESYS`

Explanation

The program specified a **`reclen`** value, and a spool file record that was longer than the program specified length was read by the library. The single long record will be split and returned to the program in segments.

Action

Reconcile the **`reclen`** value specified in the program with the length of the records being read from the JES spool.

---

539 `WARNING: File not opened, ''a+'' I/O not supported for multi-volume file.`
`Errno value: EFATTR`

Explanation

The file being opened is a multi-volume file. Update append mode (open mode **`''a+''`** or **`''a+b''`**) for a multi-volume file is not supported by SAS/C.

Action

Avoid the use of update append mode for multi-volume files.

---

539 `WARNING: Incorrect length record read from [dsname]. DIAG X'20' return code X'03'.`
`Errno value: EDEVICE`

Explanation

While reading an OS data set, the library received the indicated return code from a CP DIAG X'20' request. This indicates that an incorrect length record was read.

Action

Contact with your site's System Programming staff for assistance. Refer to *CP Programming Services* for information on DIAG X'20' return codes.

---

540 `WARNING: Permanent I/O error for device [address]. Sense bytes=[hex]`

`Errno value: EDEVICE`

Explanation

A permanent error occurred on the specified device. The message includes "sense data" which is information received from the device on the nature of the error.

Action

Consult with your site's Systems Programming Staff for information on the sense data for the failing device.

---

541 `WARNING: Device [address] [text].`
`Errno value: EDEVICE`

Explanation

A problem occurred on the device whose address is shown in the message. Possible causes include: device not attached, device halted, and device type not supported.

Action

Consult with your site's Systems Programming staff for additional help in determining the problem with the device.

---

542 `WARNING: Internal error in ''l$cdscb''. DSCB not found.`
`Errno value: ELIBERR`

Explanation

The library was searching an OS disk table of contents for the data set to be opened. The name was found; however, the DSCB was not a Format 1 DSCB. This condition should not occur.

Action

Contact SAS/C Technical Support.

---

543 `WARNING: Member name required for output to PDS.`
`Errno value: EUSAGE`

Explanation

An attempt was made to open a PDS for output without specifying a member name. When a PDS is opened for output a member name must be specified.

Action

Make sure that a member name is specified when opening an output PDS.

---

543 `WARNING: Dataset not recognized as an OS MACLIB or TXTLIB.`
`Errno value: EFATTR`

Explanation

A request was made to open a member of an OS MACLIB or TXTLIB. However, the library determined that the file was not an OS partitioned data set with RECFM FB and LRECL 80. The file was not opened.

Action

Do not attempt to process an OS data set as a MACLIB or TXTLIB unless it has the characteristics described above.

---

544 `WARNING: Dataset not found: <name>.`
`Errno value: ENFOUND`

Explanation

The library was searching for a GLOBALed OS MACLIB or TXTLIB. (An OS MACLIB or TXTLIB is considered to be GLOBALed when it is FILEDEFed using the DDname CMSLIB, and CMSLIB appears in the CMS GLOBAL list.) The data set specified was not found.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more information on GLOBALed OS MACLIBs and TXTLIBs as advanced CMS I/O facilities.

---

544 `WARNING: File not opened, no data sets could be located for external writer name ''[name]''.`
`Errno value: ENFOUND`

Explanation

An EXEC CICS SPOOLOPEN command failed with a NOTFND response code. Usually this means that no dataset with the specified name could be found.

Action

Verify that the name is correct and that the file actually exists.

---

545 `WARNING: Open failed due to physical I/O error: [text]`
`Errno value: EDEVICE`

Explanation

While opening a RECFM FBS data set for `''rel''` access, the library attempted to read the last record of the file, but a physical I/O error occurred. The [text] is information obtained from the MVS SYNADAF macro.

Action

Check for IBM messages located in the job log, such as IOS000I messages. Consult with your site's Systems Programming Staff to determine whether or not a hardware failure occurred.

---

545 `WARNING: Internal error in ''l$cggrd''. End of file not found.`
`Errno value: ELIBERR`

Explanation

While the library was reading a file the end of file indicator was not found as expected. The most probable cause is an internal library error.

Action

Contact SAS/C Technical Support.

---

546 `WARNING: Unable to reopen file. Further use is impossible.`
`Errno value: ESYS`

Explanation

A file accessed via the **''rel''** access method ran out of space. The library attempted to reopen the file after the failure, but the reopen was unsuccessful. The file can no longer be accessed, even if **clearerr** is used to attempt to clear the error.

One possible cause of this problem is use of the JCL parameter FREE=CLOSE.

Action

Check for other library messages and IBM messages located in the job log. If the problem cannot be determined, call SAS/C Technical Support.

---

546 **WARNING: Dataset not found. Filemode not defined.**
Errno value: ENFOUND

Explanation

An error occurred opening a GLOBALed OS MACLIB or TXTLIB member. The CMSLIB FILEDEF defining a GLOBALed OS MACLIB or TXTLIB did not specify a filemode, which prevents the library from locating the data set.

Action

Correct the CMSLIB FILEDEF to specify a valid filemode.

---

547 **WARNING: Internal error in ''l$catap''. Device name not found in DEVTAB.**
Errno value: ELIBERR

Explanation

The library was searching for the virtual address of an attached tape drive in the CMS device table. The end of the list was reached without locating the device. The most probable cause is an internal library error.

Action

Contact SAS/C Technical Support.

---

548 **WARNING: [access method] I/O failure: [text]**
Errno value: EDEVICE

Explanation

An operating system access method, such as BSAM, detected an error. The [text] attempts to provide more detailed information on the failure. This can be the result of a hardware problem. However, it can also be caused by misuse of I/O functions, such as attempting to read a concatenated data set where the data set with the largest blocksize is not first in the list.

Action

Check for other library messages and IBM messages located in the JES log, such as message IOS000I, which indicates a hardware failure. Contact your site's Systems Programming Staff.

---

549 **WARNING: [access method] I/O failure ([text]). Probable cause: invalid seek address.**
Errno value: EDEVICE

Explanation

An operating system access method, such as BSAM, detected an error. The [text] attempted to provide more detailed information on the failure. A seek operation preceded the error; possibly, the error may have been caused by use of an incorrect seek address.

Action

Check for errors in the arguments to **fseek** or **fsetpos**. If none are found, treat this error like message 548.

---

550 `WARNING: Record not written, file is full.`
`Errno value: ENOSPC`

Explanation

During file output, a write operation failed because these was no space left in the file, and the file could not be extended. For any access method other than **''rel''**, any further access to the file (without closing it and reopening it) is impossible. However, with **''rel''** access it is possible to continue to use the file after calling **clearerr**, as long as no attempt is made to add new records.

Action

Recreate the data set with more space, or make more free space available on the volume where the file resides. If the file is a PDS member, compressing the PDS may make more free space available.

---

551 `WARNING: File not opened, existing DCB attributes are inconsistent. Errno value: EFATTR`

Explanation

The DCB for a data set specified in JCL or in the data set label was not consistent. For instance, for a RECFM F file the LRECL did not evenly divide the BLKSIZE.

Action

Correct the JCL (or TSO allocations) to specify consistent DCB parameters.

---

552 `WARNING: A file with record format FBS cannot be opened for append. Errno value: EFATTR`

Explanation

An attempt was made to open a data set with RECFM FBS for append (open mode **`''a''`**, **`''ab''`**, **`''a+''`**, or **`''a+b''`**). Due to the characteristics of a RECFM FBS data set, it is not possible to open one for append using the **`''seq''`** access method.

Action

Use the **`''rel''`** access method, or use a data set with a different record format if append mode is required.

---

553 `WARNING: A concatenated file cannot be opened for writing. Errno value: EFATTR`

Explanation

An attempt was made to open a concatenated DDname for output. This is not supported by the operating system.

Action

Do not attempt to use a concatenated DDname for output.

---

554 `WARNING: Text I/O not supported for relative or linear VSAM file. Errno value: EFATTR`

Explanation

Linear VSAM (DIV) files can only be accessed in binary mode. Relative record (RRDS) VSAM files can be accessed only in binary or keyed mode.

Action

Correct the program to use a supported I/O mode. If text I/O is required, open some file type other than relative record or linear VSAM.

---

555 **`WARNING: File not opened, text access not supported by ''rel'' access method.`**
`Errno value: EUSAGE`

Explanation

An open was attempted that specified text access but the use of the **`''rel''`** access method was also requested. The **`''rel''`** access method is only supported for binary access.

Action

Either use an access method that supports text, or specify binary access.

---

556 **`WARNING: File not opened, option ''[option]=[value]'' incompatible with ''rel'' access method.`**
`Errno value: EUSAGE`

Explanation

A call to **`afopen`** requested the **`''rel''`** access method but also specified an amparm that is incompatible with **`''rel''`** access, such as **`trunc=yes`**. The open failed.

Action

Correct the incompatibility.

---

557 **`WARNING: Uninitialized VSAM file cannot be opened for input.`**
`Errno value: ENFOUND`

Explanation

An attempt was made to open a VSAM file with open mode **`''r+''`**, **`''r+b''`**, or **`''r+k''`**. No records had ever been stored in the file, and therefore the open failed.

This message usually occurs when a new file has been defined using Access Method Services, but no data has been written to the file.

Action

Either add a record to the file before running the program, or correct the program to use a different open mode, such as **`''w+''`** or **`''a+''`**. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for information on the meanings of open modes when used with VSAM files.

---

560 **`WARNING: Minidisk [letter] not accessed.`**
`Errno value: EUSAGE`

Explanation

The **`remove`** function has been called to remove a file on a minidisk that is not accessed.

Action

Access the minidisk, or specify the file to be removed correctly.

---

561 **`WARNING: CMS RENAME command failed with return code [num]`**
`Errno value: EUSAGE`

Explanation

The **`rename`** function invoked the CMS RENAME command that failed with the return code indicated.

Action

Refer to the *CMS Command Reference* for an explanation of the return code.

562 **`WARNING: No read/write disk accessed for ''[fname ftype *]''`**
`Errno value: EUSAGE`

Explanation

A file with filemode "*" was opened for output; however, no read/write disk was accessed, and therefore the file could not be created on any disk.

Action

Access a disk for read/write mode before running the program.

563 **`WARNING: ''[name]'' prefix is unsupported.`**
`Errno value: EARG`

Explanation

The "name" style prefix is not supported by **`cmspid`**. **`cmspid`** supports only style prefixes of **`''cms''`**, **`''xed''`**, and **`''sf''`**.

Action

Correct the program. See the *SAS/C Library Reference, Volume 1*, for more information on **`cmspid`**.

564 **`WARNING: ''[name]'' is not a valid CMS fileid.`**
`Errno value: EARG`

Explanation

The path name passed to **`cmspid`** is not composed of a valid CMS filename, filetype and filemode. The filemode is optional, but if present it must be valid.

Action

Specify a valid file name. Refer to the *CMS User's Guide* for an explanation of valid CMS fileids.

565 `WARNING: Attempt to skip over unwritten variable-length record.`
`Errno value: EUSAGE`

Explanation

The library attempted to seek past the end of a CMS RECFM V file and failed. The probable cause was the program called **`fseek`** or **`fsetpos`** specifying a file position that indicated a record past the current end of file.

Action

Correct the program. If an **`fseek`** or **`fsetpos`** argument has been overlaid with invalid data, use the debugger **`monitor`** command to locate the point of overlay.

---

566 `WARNING: Minidisk [letter] is full.`
`Errno value: ENOSPC`

Explanation

A write operation failed because the minidisk containing the output file had no more free space.

Action

Remove files from the mindisk, access another minidisk to contain the file, or change the program to write less data.

---

567 `WARNING: Attempt to seek past end of record.`
`Errno value: EUSAGE`

Explanation

An attempt has been made to seek past the end of a record. The most likely cause is the use of a seek target which is uninitialized or has been corrupted.

Action

Check that the argument passed to **`fsetpos`** or **`fseek`** is correct and has not been corrupted.

---

568 `WARNING: Attempt to seek past end of file.`
`Errno value: EUSAGE`

Explanation

While performing a **`fsetpos`** or **`fseek`** operation, the library detected the end of file. Seeking past the end of file is not supported in MVS by the **`''seq''`** access method. This condition is usually caused by the use of a seek target which is uninitialized or corrupted.

Action

Check that the argument passed to **`fsetpos`** or **`fseek`** is correct and has not been corrupted.

---

569 **NOTE: The specification ''trunc=no'' is incompatible with this type of file. ''trunc=yes'' is used.**
Errno value: unchanged

Explanation

The program requested **trunc=no** when this was not supported. For instance, **trunc=no** is incompatible with **print=yes**. The incompatible **trunc** specification is ignored.

570 **WARNING: Open failed, device does not exist.**
Errno value: ESYS

Explanation

A request was issued to open a device such as READER, PRINTER, etc. The device is not defined to CP.

Action

If the program is specifying the correct device check with your site's Systems Programming staff on why the device is unknown.

571 **WARNING: Open failed, grow=no amparm not supported with DISP=MOD.**
Errno value: EUSAGE

Explanation

An attempt was made to open a DDname defined with DISP=MOD using the amparm **grow=no**. DISP=MOD and **grow=no** are incompatible, since DISP=MOD requests adding new records to the end of the file.

Action

Correct the JCL or the application.

572 **WARNING: Open failed, grow=no amparm not supported for tape files.**
Errno value: EFATTR

Explanation

An attempt was made to open a tape file using the amparm **grow=no**. Due to limitations of magnetic tape devices, this cannot be supported.

573 **WARNING: Open failed, grow=no not supported for multi-volume spanned file.**
Errno value: EFATTR

Explanation

An open request for a multi-volume spanned format data set specified **grow=no**. Due to operating system limitations, this could not be supported, and the open was failed.

Action

Do not specify **grow=no** when opening multi-volume spanned data sets.

573 **WARNING: File [name] is not a VSAM data set.**
Errno value: ENFOUND

Explanation

An attempt was made to open a **''ddn''** style file name for keyed access, but the DDname was not defined by a DLBL command as a VSAM file.

Action

Use the DLBL command to define the VSAM file before running the program.

---

574 **WARNING: Unable to synchronize file, not positioned to end of block.**
Errno value: EUSAGE

Explanation

A call to **afflush** was made for a multi-volume file opened with **grow=no**. Due to operating system limitations, this is only permitted when the file is positioned to the end of a block.

Action

Avoid the use of **afflush** for multi-volume files opened with **grow=no**.

---

575 **WARNING: File position unknown due to previous error.**
Errno value: EPREV

Explanation

A call was made to **ftell** or **fgetpos**, but a previous error made determination of the current file position impossible. A diagnostic should have been generated describing the previous error, which was probably associated with the errno value **EDEVICE**.

Action

Use **fseek**, **fsetpos**, or **rewind** to define a valid file position after an **EDEVICE** error.

---

576 **WARNING: afflush not supported for PDSE member opened with ''grow=yes''.**
Errno value: EUSAGE

Explanation

Due to PDSE restrictions, the **afflush** function was not implemented for PDSE members opened with the amparm **grow=no**.

Action

If **afflush** is a requirement, consider using a standard PDS in place of a PDSE.

577 **WARNING: afflush not supported by ''seq'' access method for RECFM FBS files.**
Errno value: EFATTR

Explanation

A call to **afflush** was made for a RECFM FBS file accessed via the **''seq''** access method. **afflush** is not supported in this case.

Action

If possible, use the **''rel''** access method, which supports RECFM FBS files and allows **afflush**. Alternately, use a RECFM F file.

---

578 **WARNING: afflush not supported for DSORG DA file.**
Errno value: EUSAGE

Explanation

A call to **afflush** was made for a DSORG DA file accessed via the **''seq''** access method. This is not supported.

Action

Remove the DSORG DA attribute from the file. Note, in general using DSORG DA with files read or written by SAS/C programs should be avoided. Use of DSORG DA adds no additional capabilities and results in additional restrictions such as the above.

---

580 **WARNING: System routine ''[name]'' failed with reason code [num]**
Errno value: ESYS/ELIBERR

Explanation

The named system routine failed with an unusual reason code. Some of the valid system routines are: DMSEXIST, DMSRENAM, DMSGETWU, DMSOPEN, DMSREAD, DMSWRITE, DMSCOMM, DMSOPDIR, and DMSGETDI.

Action

Refer to the *CMS Application Development Reference* for information on the reason code. The reason code should identify the cause of the problem and an appropriate response.

---

581 **WARNING: File not opened, ''org=[value]'' was specified, but actual organization is ''[value]''**
Errno value: EFATTR

Explanation

A call to **afopen** specified the **org** amparm, but the actual organization of the file to be opened was incompatible. The file was not opened.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more information on access method parameters.

---

582 **WARNING: File not opened. ''org=[value]'' was specified, but no member name was defined.**
Errno value: EFATTR

Explanation

An **afopen** call specified one of the amparms **org=po, org=pds** or **org=pdse**. However, a member name was not specified either in the file name or by the DD statement or TSO allocation. The open failed.

Action

Do not specify partitioned organization for a file without a member name. If you want to read a PDS directory, specify **org=ps**.

---

583 **WARNING: This file cannot be opened for keyed access.**
Errno value: EFATTR

Explanation

A request has been made to open a file for keyed access that does not support it. Only VSAM KSDS, ESDS, and RRDS files can be opened for keyed access.

Action

Specify a valid VSAM data set name, or correct the program not to use keyed access.

---

585 **WARNING: Internal error in [name]: Unexpected value for [text].**
Errno value: ELIBERR

Explanation

While checking the access mode of a CMS shared file system file, the named library routine found the authority flags to be blank or invalid. This is a library internal error.

Action

Contact SAS/C Technical Support.

---

586 **WARNING: [name] is not a valid ``[type]'' style pathname.**
Errno value: EARG

Explanation

The name specified is not a valid pathname for the style prefix specified.

Action

Correct the program. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for complete details.

---

587 `WARNING: Invalid file identifier: [name].`
`Errno value: EARG`

Explanation

The specified file identifier is invalid. The filename, filetype or filemode does not conform to CMS standards.

Action

Correct the program. See the *CMS User's Guide* for information on valid file name components.

---

588 `WARNING: Attempt to use the default file pool, which is currently undefined.`
`Errno value: EARG`

Explanation

The program attempted to use the default file pool, but no default file pool was defined.

Action

To define a default file pool, use the CMS command SET FILEPOOL. Refer to the *CMS User's Guide* for usage of the SET FILEPOOL command.

---

589 `WARNING: Unable to obtain a work unit ID.`
`Errno value: ELIBERR`

Explanation

The library attempted to fill in a Open Work Area but the request failed.

Action

This message will be accompanied by message 580. Refer to that message for more information.

---

590 `WARNING: CMS shared files can only be accessed using the ''rel'' or ''seq'' access methods.`
`Errno value: EFATTR`

Explanation

The program attempted to access a CMS shared file specifying an access method other than `''rel''` or `''seq''`. This is unsupported.

---

591 **`WARNING: CMS shared file directories can only be accessed using the ''seq'' access method.`**
`Errno value: EFATTR`

Explanation

CMS shared file directories (**`''sfd''`** style pathnames) must be accessed with the **`''seq''`** access method.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more details on CMS shared files.

---

592 **`WARNING: File [name] cannot be positioned to record number [num].`**
`Errno value: EUSAGE`

Explanation

A request was made to reposition the indicated file. When the library attempted to position the file it determined that the position requested was not valid.

Action

Make sure the file position is valid. If an **`fseek`** or **`fsetpos`** argument has become corrupted, consider using the SAS/C debugger **`monitor`** command to locate the problem. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for details on file positioning in CMS.

---

593 **`WARNING: CMS shared file [name] not found.`**
`Errno value: ENFOUND`

Explanation

The specified CMS shared file was not found in any of the defined directories.

---

594 **`WARNING: File or directory ''[name]'' not found or you are not authorized to [action] it.`**
`Errno value: ENFOUND`

Explanation

A request was made to perform an action such as rename or remove on the specified directory. The directory does not exist, or the caller is not authorized to perform the action.

Action

Specify the name correctly, and do not attempt to remove or rename files for which you are not authorized.

---

595 **`WARNING: Open failed, CMS shared file directory [name] not found.`**
`Errno value: ENFOUND`

Explanation

An open request for a file in a CMS shared file directory or for the directory itself failed because the directory was not found.

Action

Correct the program or file name.

---

596 **`WARNING: Open failed, CMS shared file directory [name] is already open.`**
`Errno value: EFFORM`

Explanation

The program attempted to open the same CMS shared file directory more than once. This is not permitted by some releases of CMS.

---

597 **`NOTE: Unsupported value specified for the dirsearch amparm, dirsearch=allauth will be used.`**

Explanation

The supported values for the amparm **`dirsearch`** are: **`file`**, **`all`**, **`allauth`**, and **`subdir`**. Some other value was specified.

Action

Correct the program. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more details on CMS shared files.

---

598 **`WARNING: Open failed, a fileid must be specified with dirsearch=file, all, or allauth.`**
`Errno value: EUSAGE`

Explanation

When using the amparm **`dirsearch`** a fileid must be specified. For instance, "sfd: * exec *" is an invalid file name with **`dirsearch=file`**.

Action

Correct the program. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more details on CMS shared files.

---

599 `WARNING: Open failed, a fileid cannot be specified with dirsearch=subdir.`
`Errno value: EUSAGE`

Explanation

When using the amparm **`dirsearch=subdir`**, the file name can specify only the directory to search. For instance, **`''sfd: master''`**.

Action

Correct the program. Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for more details on CMS shared files.

---

600 `WARNING: File not opened, blksize larger than track size.`
`Errno value: EUSAGE`

Explanation

An attempt was made to open a file, but the **`blksize`** amparm specified a block size larger than the track size of the device on which the file was allocated.

Action

Request a smaller block size, or change your allocation procedures to use a device with a larger track size.

---

601 `WARNING: File not opened, ''[name]'' is not a proper temporary filename.`
`Errno value: EARG`

Explanation

An internal error occurred in library temporary file processing.

Action

Contact SAS/C Technical Support.

---

602 `WARNING: Temporary file not opened, no [name] DD card.`
`Errno value: ENFOUND`

Explanation

In systems without dynamic allocation such as VS1, a SYSTMP[nn] DD statement is required for each temporary file opened by the program. On these systems, if no DD statement is defined, the open fails.

Action

Add appropriate DD statements and run the job again.

---

603 **`WARNING: Unable to create temporary file, DYNALLOC macro returned error code [num], reason code [hex].`**
`Errno value: ESYS`

Explanation

An attempt to dynamically allocate a temporary file failed. The error code and reason code for the failure are shown in the message.

Action

Look up the codes in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs* publication and take appropriate action.

---

604 **`WARNING: Temporary file not created, sufficient disk space not available.`**
`Errno value: ENOSPC`

Explanation

Dynamic allocation failed to create a temporary file because there was not enough disk space available.

Action

Report this problem to your site's Systems Programming Staff. On CMS, this problem can be addressed by accessing a temporary disk.

---

605 **`WARNING: File not opened, ddname ''[name]'' does not define a temporary disk file.`**
`Errno value: EFATTR`

Explanation

The library attempted to open a temporary file defined by a SYSTMP[nn] DD statement. The DD statement did not define a temporary disk data set.

Action

Correct or remove the DD statement.

---

606 **`WARNING: The ''rel'' access method does not support DISP=MOD.`**
`Errno value: EFATTR`

Explanation

An attempt was made to open a file defined with DISP=MOD using the **`''rel''`** access method, which does not support DISP=MOD.

Action

Consider replacing the use of DISP=MOD with open mode **`''ab''`** or **`''a+b''`**. Alternately, change the file's RECFM to FB from FBS, and specify use of the **`''seq''`** access method, which can support DISP=MOD.

---

607 **`WARNING: Temporary file not created, unit name not defined: [name]`**
`Errno value: ESYS`

Explanation

An attempt to dynamically allocate a temporary file failed because the unit name requested for the temporary file was not known to the system. The unit name for temporary files is VIO by default; however, the unit name can be changed by means of a zap described in the SAS/C installation instructions.

Action

See the installation instructions for information on changing this unit name, or contact SAS/C Technical Support.

---

608 **`WARNING: Temporary file not created, ddname [name] already in use.`**
`Errno value: EINUSE`

Explanation

This message normally indicates an internal library error in opening a temporary file. The message can also be produced if multiple tasks in an address space are executing the SAS/C debugger, and a SYSTMPDB DD statement has been defined.

Action

If you are running the debugger for more than one task, either avoid using a SYSTMPDB DD statement or restrict use of the debugger to a single task. Otherwise, contact SAS/C Technical Support.

---

609 **`WARNING: I/O operation failed. Probable cause: no more extents can be allocated.`**
`Errno value: ENOSPC`

Explanation

While processing a linear data set, the library issued the DIV macro. The DIV macro failed with reason code X'E10802', indicating an attempt to map beyond the end of file. This code usually indicates that it was not possible to allocate a new extent for the file.

Action

Recreate the data set with enough space. If the cause does not appear to be lack of disk space, contact SAS/C Technical Support.

---

610 **`WARNING: Specified bufsize extends past last extent of file.`**
`Errno value: EFATTR`

Explanation

An attempt was made to open a linear data set specifying a **`bufsize`** amparm which extended past the last extent of the file. The open failed.

Action

Either decrease the **bufsize** specification, or recreate the file with more space.

611 **NOTE: Linear Dataset not allocated with SHAREOPTIONS(1,3).**
Errno value: unchanged

Explanation

An attempt was made to open a linear data set (LDS) defined with a SHAREOPTIONS value other than SHAREOPTIONS(1,3). Correct processing of a linear data set cannot be guaranteed if the data set is shared. This note is generated to inform the user; however, processing continues.

Action

Since the results of sharing an open linear data set are unpredictable, consider redefining the file as SHAREOPTIONS(1,3). To suppress the message, see the description of the **quiet** function in the *SAS/C Library Reference, volume 1.*

640 **WARNING: A [filetype] file cannot be [operation].**
Errno value: EFATTR

Explanation

An attempt was made to remove or rename a terminal file. This is not supported.

644 **WARNING: Virtual machine must be in 370-mode to perform this function.**
Errno value: EFORBID

Explanation

An attempt was made to open a **%MACLIB** or **%TXTLIB** file where the GLOBAL list identified a file on an OS disk. CMS was IPLed in XA mode, but this form of access to OS disks is supported only in 370 mode.

Action

IPL CMS in 370 mode, or access the OS data set via a specific FILEDEF rather than as a GLOBALed MACLIB or TXTLIB.

645 **WARNING: File not opened, authorization is required to [operation] this file.**
Errno value: EACCES

Explanation

An attempt to open a VSAM file failed because the user did not have appropriate authorization. [operation] specified the level of access required.

Action

If the access is appropriate, use your site's security program to permit the access.

646 **`WARNING: File not opened due to simultaneous incompatible use by other users.`**
`Errno value: EINUSE`

Explanation

An attempt to open a VSAM data set failed because the data set was in use by another job or user. This message will be generated even if all users have the data set allocated with DISP SHR unless sharing is permitted by the data set's SHAREOPTIONS. Note, however, that SAS/C does not support sharing of VSAM files by multiple users. If the SHAREOPTIONS allow sharing, the library will not detect when sharing occurs, and the results will be unpredictable.

Action

Run the program when no other users are utilizing the data set.

---

647 **`WARNING: Open failed, output file already exists and is not reusable.`**
`Errno value: EFATTR`

Explanation

An attempt was made to open and truncate an existing VSAM file using an open mode beginning with **`'w'`**, such as **`''wb''`**. The file was not defined by Access Method Services as REUSABLE, and therefore the file could not be opened as specified.

Action

Use Access Method Service to make the file REUSABLE, or use an open mode that does not attempt to truncate the file (for example, **`''r+b''`**).

---

648 **`WARNING: VSAM [name] macro failed, record held by another program or file.`**
`Errno value: EINUSE`

Explanation

An attempt was made to update or delete a record of a VSAM data set, but the control interval containing the record is held by another **`FILE`** pointer referencing the same data set, or by another task in the same address space. The other **`FILE`** or task must release control of the interval before the request will be allowed.

Action

Refer to "I/O Functions" in *SAS/C Library Reference, Volume I*, for information on how to code the program to deal with this condition.

---

649 **`WARNING: VSAM file can no longer be accessed due to reopen failure. Errno value: ESYS`**

Explanation

The library had to close and reopen a VSAM file to switch from "load mode" to "update mode." The reopen request failed, and therefore the file is no longer accessible. Using the JCL option FREE=CLOSE can cause this error.

Action

If FREE=CLOSE was specified, remove it. Otherwise, check the job log for IBM messages describing the reason the open failed, and take appropriate action.

---

650 **`WARNING: VSAM [name] macro failed with return code [num]. Errno value: ELIBERR/ESYS`**

Explanation

A VSAM macro failed with the indicated return code. If the **`errno`** is **`ELIBERR`**, it indicates a condition the library believes should not occur. If the **`errno`** is **`ESYS`**, it indicates a failure which could be either an environmental error or a library error.

Action

If the **`errno`** is **`ELIBERR`**, contact SAS/C Technical Support. If the **`errno`** is **`ESYS`**, look up the macro in *MVS/ESA Macro Instructions for Data Sets* and take the action indicated for the return code.

---

651 **`WARNING: VSAM [operation] failed with reason code [num]. Errno value: ELIBERR/ESYS`**

Explanation

A VSAM operation failed, generating this reason code. If the **`errno`** is **`ELIBERR`**, it indicates a condition the library believes should not occur. If the **`errno`** is **`ESYS`**, it indicates a failure which could be either an environmental error or a library error.

Action

If the **`errno`** is **`ELIBERR`**, contact SAS/C Technical Support. If the **`errno`** is **`ESYS`**, look up the macro in *MVS/ESA Macro Instructions for Data Sets* and take the action indicated for that return code. (For the [operation] of "temporary close," see the CLOSE macro description.)

---

652 **`WARNING: VSAM [name] macro failed due to insufficient memory. Errno value: ENOMEM`**

Explanation

A **`VSAM`** macro failed because of an out-of-memory condition.

Action

Specify a larger region or virtual machine size, or reduce the program's memory requirements.

---

653 **`WARNING: I/O request failed due to hardware error during execution of the [name] macro.`**
`Errno value: EDEVICE`

Explanation

An I/O request failed due to a hardware error.

Action

In MVS, consult the SYNADAF information (sent to the MVS console in message 098) for more information. Report this error to your site's Systems Programming Staff.

---

654 **`WARNING: VSAM [name] macro failed with a logical error, reason code [num].`**
`Errno value: ELIBERR/ESYS`

Explanation

An unexpected error was indicated by the VSAM [name] macro during an I/O operation. The file's error flag is set. **`errno`** is set to **`ELIBERR`** if the error indicates a library failure.

Action

Look up the reason code in the *MVS/ESA Macro Instructions for Data Sets* publication, and take appropriate action. If **`errno`** is **`ELIBERR`**, contact SAS/C Technical Support.

---

655 **`WARNING: VSAM [operation] not performed to due to previous seek failure.`**
`Errno value: EPREV`

Explanation

The VSAM operation failed a previous seek request failed.

Action

Refer to the message(s) describing the seek failure for further information.

---

656 **`WARNING: Seek failed, target does not address a valid record.`**
`Errno value: EUSAGE`

Explanation

A VSAM seek request failed because the target position did not specify the RBA of any record. This may have been caused by overlaying the argument to **`fseek`** or **`fsetpos`**.

Action

Verify that the seek address is correct and has not been overlaid. If the seek address has been overlaid, use the SAS/C debugger **`monitor`** to determine the point of overlay.

---

657 **`WARNING: File not opened, unable to locate end of file.`**
`Errno value: ELIBERR`

Explanation

The library was unable to locate the last record of the VSAM data set being opened.

Action

Contact SAS/C Technical Support.

---

658 **`WARNING: A VSAM KSDS cannot be opened for output in text or binary mode.`**
`Errno value: EARG/EFATTR`

Explanation

The user attempted to open a KSDS file for output or update in text or binary mode. This is not supported. The **`errno`** variable is set to **`EARG`** if the amparm **`org=ks`** is specified; otherwise, the **`errno`** is set to **`EFATTR`**.

Action

Modify the program to use keyed access, or open a file of some type other than a KSDS.

---

660 **`WARNING: File not opened, specification ``[keyword]=[value]'' conflicts with actual value of [value].`**
`Errno value: EFATTR`

Explanation

An attempt was made to open a file where the **`keylen`** or **`keyoff`** amparm did not agree with that of the file to be opened.

Action

Ensure that files opened by the program conform to the **`keylen`** and **`keyoff`** specifications within the program.

---

661 **`WARNING: This file does not permit the length of a record to be changed.`**
`Errno value: EUNSUPP`

Explanation

The program attempted to replace a record in a VSAM ESDS or RRDS with one whose length was different. Changing the length of a record with **`kreplace`** is supported only for a KSDS. Note, that even though an alternate index over an ESDS can be treated as a KSDS for most purposes, these files do not support updates that change the length of a record.

Action

Correct the program to not modify the lengths of records, or ensure that it always opens a KSDS.

---

662 `WARNING: Retrieval failed, file position undefined due to previous error.`
`Errno value: EPREV`

Explanation

Due to a previous error in VSAM processing, the file position in the cluster is unknown. For this reason, it was impossible to retrieve the next sequential record.

Action

Check the message for the previous error and take appropriate action.

---

663 `WARNING: Record has duplicate key, not inserted.`
`Errno value: EDUPKEY`

Explanation

An attempt was made to add a record with the same key as another record in the file, and the file does not support this. The error flag is set for the file.

Action

Don't attempt to add a duplicate record unless the file is an alternate path that allows duplicate keys. Note that this message is suppressed by default, and therefore is generated only when the run-time option **`=warning`** is in effect.

---

664 `WARNING: Request failed due to invalid relative record number: [num].`
`Errno value: EUSAGE`

Explanation

The key specified for a record in a RRDS is not a valid record number for this file.

Action

Correct the program to specify a valid record number as the record key.

---

670 `WARNING: File [text], DYNALLOC macro returned error code [num], reason code [hex].`
`Errno value: ESYS`

Explanation

An attempt to dynamically allocate a data set failed. If the error occurred in **`osdfind`** or **`osdnext`**, [text] gives the name of the file which could not be allocated. If the error occurred in some other function, the [text] describes the operation being performed.

Action

Look up the error codes in the IBM *MVS/ESA Application Development Guide: Authorized Assembler Language Programs*, and take appropriate action. If the error codes indicate a programming problem, this is a SAS/C library internal error.

---

670 **`WARNING: An OS format file cannot be accessed with the [type] style prefix.`**
`Errno value: EUNSUPP`

Explanation

The library does not support access to filemode 4 (OS simulated) files using an **`sf`** or **`sfd`** style file name.

Action

ACCESS the shared file directory as a minidisk, and open the file using a **`cms`** style file name.

---

671 **`WARNING: member not [operation], ddname does not define a partitioned data set.`**
`Errno value: EFATTR`

Explanation

An attempt to remove or rename a PDS member failed because the DDname specified did not point to a PDS.

Action

Make sure the DDname is defined as a PDS (or PDSE).

---

672 **`WARNING: [text], file could not be opened.`**
`Errno value: ESYS`

Explanation

The operation indicated by [text] failed because the file to be modified or checked could not be opened.

Action

Check the job log for IBM messages describing the cause of the failure, and take appropriate action.

---

673 **`WARNING: [text], ddname does not define a physical sequential file.`**
`Errno value: EFATTR`

Explanation

The **`remove`** function called for a DDname that identified a file whose organization was not physical sequential. For a DDname that is not a PDS member, **`remove`** attempts to empty the file, but because of the file's organization, this was not possible.

Action

Do not attempt to call **`remove`** for an unsupported file type.

---

674 `WARNING: [text], member not found.`
`Errno value: ENFOUND`

Explanation

An attempt to remove or rename a PDS member failed because the PDS member did not exist.

Action

Specify an existing member name.

---

675 `WARNING: Member not renamed, old ddname ''[name]'' does not match new ddname ''[name]''.`
`Errno value: EARG`

Explanation

A call to **`rename`** attempted to rename a member of one DDname to a member of another DDname. This operation is not supported.

Action

Correct the program to specify the same DDname for both the old and new member.

---

676 `WARNING: [text], new member name already exists.`
`Errno value: EUSAGE`

Explanation

An attempt to rename a member of a PDS failed because the new member name was already defined.

Action

Specify a member name which does not already exist.

---

679 `WARNING: Insufficient authority to [operation] this file.`
`Errno value: EACCES`

Explanation

A call to **`remove`** or **`rename`** failed because the user was not authorized to perform this operation on the file.

Action

Correct the file name, or use your site's security package to get authorization for the file.

---

680 **`WARNING: A [filetype] can not be renamed to a [filetype].`**
`Errno value: EUSAGE/EFATTR`

Explanation

A call to **`rename`** attempted to rename one kind of data set as another kind. For example, the old name specified a PDS member, but the new name did not. Renames which change the type of a data set are not supported. The **`errno`** is **`EUSAGE`** if the error was in the **`rename`** arguments, or **`EFATTR`** if the error depended on the environment (e.g., occurred because a DDname was allocated to an HFS file).

Action

Correct the program or the file definitions to make the types compatible.

---

681 **`WARNING: Old PDS name and new PDS name must be the same to rename a PDS member.`**
`Errno value: EUSAGE`

Explanation

A call to **`rename`** requested renaming a PDS member to be a member of a different PDS. This operation is not supported. A rename involving a PDS member must change only the member name.

---

682 **`WARNING: File not removed, expiration date not passed.`**
`Errno value: ESYS`

Explanation

An attempt to remove a sequential data set failed because its expiration date had not yet passed.

Action

Use a system utility such as IEHPROGM to remove the data set.

---

683 **`WARNING: File not renamed, can not rename to a generation data group name.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to rename a data set to a member of a generation data group. This is not supported.

Action

Correct the program. For more information on generation data groups, refer to the IBM manual *MVS/DFP Using Data Sets.*

---

684 **`WARNING: [operation] failed, [new] DDname references a HFS file.`**
`Errno value: EUSAGE`

Explanation

A call was made to **`remove`** or **`rename`** specifying a DDname that referenced an HFS file. **`remove`** and **`rename`** are not supported with DDnames allocated to HFS files unless the "ddname/filename" syntax is in use.

Action

Do not attempt to call **`remove`** or **`rename`** for DD statements addressing HFS files.

---

690 **`WARNING: File not found.`**
`Errno value: ENFOUND`

Explanation

A call to **`remove`** or **`rename`** specified a non-existent data set name.

Action

Correct the file name, or check for its existence using **`access`** before calling **`remove`** or **`rename`**. To suppress the message, use the **`quiet`** function as described in the *SAS/C Library Reference, Volume 1.*

---

691 **`WARNING: Tape file can not be renamed.`**
`Errno value: EFATTR`

Explanation

An attempt has been made to rename a tape data set. This is not supported.

Action

Correct the program or file name.

---

692 **`WARNING: File name is longer than 44 characters.`**
`Errno value: EARG`

Explanation

The argument to **`remove`** or **`rename`** specified a member of a generation data group. However, when the name was expanded, it was longer than 44 characters, which is the MVS limit for data set names.

Action

Specify a valid file name.

---

693 `WARNING: File not renamed, new name already exists on volume ''[name]''.`
`Errno value: EUSAGE`

Explanation

A rename operation failed because the new name already existed on the volume specified.

Action

When calling **rename**, make sure the new file name does not already exist.

---

694 `WARNING: File not renamed, could not be found on volume ''[name]''`
`Errno value: ENFOUND`

Explanation

An attempt to rename a file failed because the specified file could not be found on the volume shown. This error probably indicates that the file to be renamed is not catalogued correctly.

Action

Correct the catalog information for the file and try again.

---

695 `WARNING: [text], data set is open.`
`Errno value: EINUSE`

Explanation

A call to **remove** or **rename** failed because the data set was open.

Action

Close the data set before attempting to remove or rename it.

---

696 `WARNING: [text], system macro ''[name]'' returned status code [num] for volume ''[name]''.`
`Errno value: ESYS`

Explanation

During a call to **remove** or **rename**, the indicated system macro produced an unexpected status code. The **remove** or **rename** operation failed.

Action

Look up the status code in the IBM *MVS/DFP System Programming Reference* for a description of the error, and take appropriate action.

---

697 `WARNING: File has not been renamed on all volumes and not recataloged.`
`Errno value: ESYS`

Explanation

An attempt to rename a multi-volume data set was successful on some volumes but not on others. Each error was described by a previous message. The library did not attempt to change the catalog or repair the damage. This must be done by the user or by Systems Programming Staff.

Action

Locate messages for the previous error(s) and take appropriate action. Consult your site's Systems Programming Staff for assistance in correcting the catalog information.

---

698 `WARNING: [name] function not supported for VSAM files.`
`Errno value: EUSAGE`

Explanation

The **`remove`** and **`rename`** function are not supported for VSAM files.

Action

VSAM files can be removed or renamed using Access Method Services.

---

699 `WARNING: System macro ''CATALOG UNCAT'' failed with return code: 8, R0: [hex], R1: [hex].`
`Errno value: ESYS`

Explanation

While removing or renaming an MVS data set, the library attempted to uncatalog the data set. The uncatalog operation failed with a system error. In the case of **`remove`**, the data set was scratched from the disk before uncatalog was attempted. In the case of **`rename`**, the library attempted to restore the file's original name after an uncatalog error occurred.

Action

Look up the failure codes in the IBM manual *MVS/DFP System Programming Reference*, and take appropriate action. If the failure indicates a programming error, contact SAS/C Technical Support.

---

700 `ERROR: System ABEND [code] occurred loading module: [name].`
`Errno value: ESYS`

Explanation

An attempt was made to load a dynamic load module, but an unexpected system error occurred which prevented the module from being loaded.

Action

Look up the ABEND code in *MVS/ESA System Codes* and take appropriate action.

---

701 `WARNING: Load module not found: [name].`
`Errno value: ENFOUND`

Explanation

An attempt was made to load a dynamic module, but the module was not found in any appropriate library. In MVS, the library searches first for the module in any active tasklibs, then in STEPLIB and JOBLIB, and finally in the system linklist and LPA. If the module appears to be a library module (that is, its name begins with LSC or L$), the library also searches the CTRANS data set.

In CMS, the library searches DYNAMC LOADLIB, as well as any other locations specified by a call to **addsrch**. For run-time library modules, it searches in LSCRTL LOADLIB on an accessed disk, or possibly in a run-time library shared segment. For more information on the CMS search order, see the *SAS/C Library Reference, Volume 2.*

Action

Confirm that the module name and search order were specified correctly. In MVS, verify that the correct libraries were specified as STEPLIB or JOBLIB in the JCL. (If running under the OpenEdition shell, verify that the **STEPLIB** environment variable has been defined appropriately.) In CMS, verify that the mini-disk containing any required LOADLIB was correctly ACCESSed. Refer to Chapter 1, "Dynamic-Loading Functions" in *SAS/C Library Reference, Volume 2*, for complete details.

---

702 **WARNING: No memory available to load module: [name].**
Errno value: ENOMEM

Explanation

An attempt was made to load a dynamic load module, but there was not enough memory to load it.

Action

Increase the program's region size, or reduce the program's memory requirements. If the module is marked AMODE=24, consider whether the application can be modified to run AMODE=31,RMODE=ANY.

---

704 **WARNING: Load module [name] marked not executable.**
Errno value: ENFOUND

Explanation

An attempt was made to load a dynamic load module that was marked not executable. This indicates that when the module was link-edited, one or more errors occurred that precluded load module execution. (A return code of 8 or greater is returned by the linkage editor when this occurs.)

Action

Verify that the load module that could not be loaded was linked successfully, with a return code less than 8. If it appears that the linkedit was successful, verify that libraries are specified in the right order in the JCL so that the module is being fetched from the correct library. If the link edit failed because routines that are not actually required for execution were absent, consider using the LET linkage editor option or the C __weak keyword.

---

705 **WARNING: Search element type [num] unknown or not supported by [system].**
Errno value: EARG

Explanation

The **addsrch** function was called to add a specification to the dynamic loading search order, but the element type was unknown or not supported by the operating system.

Action

Verify that the search element type is one of the symbolic constants defined for this purpose in **<dynam.h>**, and that the element type is appropriate to the operating system. See *SAS/C Library Reference, Volume 2*, for details. Note that under MVS calls to **addsrch** specifying a CMS search element type are ignored, and no diagnostic is generated.

---

708 **NOTE: Attempt to delete undefined search pointer.**
Errno value: unchanged

Explanation

The function **delsrch** was called to delete a search element from the search list. The search element was not valid or has already been deleted. This message may indicate that the search element has been overlaid.

Action

Verify that the search element was stored correctly and has not been overlaid. If an overlay has apparently occurred, use the SAS/C debugger command **monitor** to locate the point of the overlay.

---

709 **WARNING: Time of day unavailable, system clock not operational.**
Errno value: EDEVICE

Explanation

A call was made to the **time** function to return the time of day. Time of day information was not available because the system time of day clock was not functional.

Action

This is most likely a hardware error. Contact your site's Systems Programming staff.

---

710 **`NOTE: Unable to determine time zone, Greenwich time assumed.`**
`Errno value: unchanged`

Explanation

One of the timing functions such as **`localtime`** attempted to determine the difference between local time and Greenwich time. This attempt failed, probably because of a hardware time-of-day clock failure. The library assumes that local time and Greenwich time should be the same.

Action

This is most likely a hardware error. Contact your site's Systems Programming staff.

---

711 **`WARNING: Argument to gmtime/localtime is not a proper time value.`**
`Errno value: EARG`

Explanation

The argument passed to the **`gmtime`** or **`localtime`** function specified a time that is outside the range of times that can be represented by the 370 time-of-day clock (from the year 1900 to the year 2041).

Action

Correct the program to pass a valid **`time_t`** value.

---

712 **`WARNING: An internal error occurred during gmtime/localtime processing.`**
`Errno value: ELIBERR`

Explanation

An internal error occurred converting a **`time_t`** value to a Gregorian date.

Action

Contact SAS/C Technical Support.

---

713 **`WARNING: Unknown system command type: [text].`**
`Errno value: EARG`

Explanation

The program attempted to invoke a system command with an unsupported style prefix. The valid prefixes in MVS are **`''PGM''`**, **`''TSO''`**, and **`''SH''`**. The valid prefixes in CMS are **`''CMS''`**, **`''CP''`**, **`''SUBSET''`**, and **`''XEDIT''`**.

Action

Specify a valid system command prefix. If the system command to be executed has a colon in its first nine characters, add the default prefix to the start of the command to keep the text before the colon from being interpreted as a prefix.

---

714 `WARNING: No command name present in ''system'' argument string.`
`Errno value: EARG`

Explanation

The argument string passed to the **system** function was entirely blank or contained only a style prefix.

Action

Correct the program to specify a command name.

---

715 `WARNING: [Command/Program] name too long: [name]....`
`Errno value: EARG/EUSAGE`

Explanation

A call to system attempted to invoke a program or command longer than 8 characters, which is the maximum supported by MVS. Note that the error may be in a CLIST or REXX EXEC invoked by the **system** function as well as in the argument to **system**.

Action

Correct the program, CLIST or REXX EXEC.

---

716 `WARNING: [Program/Command] [name] was abnormally terminated with a [system/user] code of [code]`
`Errno value: ESYS`

Explanation

A program or TSO command invoked by the **system** or **oslink** function, or by one of the MVS SUBCOM functions, abnormally terminated. In the case of **system**, the failing command may have been specified by a CLIST or REXX EXEC rather than by the caller of **system**.

Action

This message generally indicates a problem with the program or command being invoked, not with the caller of **oslink** or **system**. For a system code, refer to *MVS/ESA System Codes* for further information on the ABEND. If the program being invoked is a SAS/C program and the ABEND code is in the range 1200–1240, see "ABEND codes" in this book for further information. For other user ABEND Codes, see the documentation for the program being invoked, or for other products used by that program (e.g., FORTRAN, DL/I).

---

717 `WARNING: Invalid TSO command name rejected: [name]`
`Errno value: EUSAGE`

Explanation

An invocation of the **system** function to run a TSO command detected an invalid TSO command name. The invalid nanme may have been specified in a TSO CLIST or REXX EXEC rather than by the argument to **system**.

Action

Correct the program, CLIST or REXX EXEC. See the *TSO/E User's Guide* for information on TSO command syntax.

---

718 `WARNING: TSO environment not active, ''system'' command rejected. Errno value: EUSAGE`

Explanation

The **`system`** function was called to issue a TSO command but the program was not running in a TSO environment. The request was rejected.

Action

Avoid use of the **`system`** function using the **`''TSO''`** style prefix for programs not run under TSO. The **`envname`** function can be used to determine if the program is running under TSO.

---

719 `NOTE: [name] signal not supported in [environment]. Errno value: unchanged`

Explanation

An attempt was made to handle a signal that is not supported in the current environment, for example, SIGINT in MVS batch or CICS. The call to **`signal`** or **`sigaction`** indicates success, but the only way the signal can be generated is using the **`raise`** function.

Action

Correct the program so that it does not attempt to handle signals that are not supported by the expected environment. Also, see the *SAS/C Library Reference, Volume 1*, for information on using the **`quiet`** function to suppress unwanted run-time messages.

---

720 `NOTE: Invalid data in TSO environment file: [line] Errno value: unchanged`

Explanation

Invalid data was found in the file containing definitions of PERMANENT scope TSO environment variables. The invalid data was ignored, and processing of the file continued.

Action

If you have edited or modified the PERMANENT scope environment variable file (normally called C@ENV.PERM), correct the file. Otherwise, contact SAS/C Technical Support. See the *SAS/C Library Reference, Volume 1*, for information on the contents of this file.

---

721 `NOTE: Unexpected end of TSO environment file.`
`Errno value: unchanged`

Explanation

An environment variable function such as **getenv** encountered an unexpected end-of-file trying to process the TSO permanent environment variable file (normally called C@ENV.PERM). Any incomplete data was ignored.

Action

Same as for message 720.

---

722 `WARNING: Unable to allocate TSO environment file, DYNALLOC error code [hex]`
`Errno value: ESYS`

Explanation

The library failed to allocate the TSO PERMANENT scope environment variable file. For a call to **getenv**, the library assumes there are no PERMANENT scope variables. For a **putenv** or **setenv** call, an error is indicated to the caller.

Action

Look up the DYNALLOC error code in *MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs* and take appropriate action. If the code indicates a programming problem, contact SAS/C Technical Support. For information on how the library allocates and uses the environment variable file, see the *SAS/C Library Reference, Volume 1.*

---

723 `WARNING: Error reading TSO environment file: [text]`
`Errno value: EDEVICE`

Explanation

An I/O error occurred reading the TSO permanent scope environment variable file. The cause (as supplied by the SYNADAF macro) is given in the message.

Action

This may be caused by allocating a file to the DDname C@ENV that does not have an appropriate DCB (RECFM VB, LRECL 259). Check the file's DCB, and also check the MVS console log for message IOS000I, indicating a hardware problem. If a hardware problem is indicated, report the message to your site's Systems Programming Staff.

---

723 `WARNING: EXEC CICS [name] TS QUEUE([name]) failed with response code [num] while attempting to access environment variables.`
`Errno value: EACCES, EDEVICE, ESYS, ENOSPC`

Explanation

An EXEC CICS READQ or WRITEQ command failed while trying to read or write the temporary storage queue used to store environment variables.

Action

The response code should indicate the action to take. Response codes can be found in the appropriate CICS documentation under EIBRESP and are listed after the READQ and WRITEQ command descriptions. For example, a response code corresponding to NOSPACE occurs when insufficient space is available in the temporary storage data set, while a response code corresponding to IOERR occurs when there is an unrecoverable I/O error in the temporary storage data set.

---

724 **`WARNING: Permanent environment variables not supported in batch. Errno value: EUNSUPP`**

Explanation

The **`putenv`** or **`setenv`** function was called to update a PERMANENT scope environment variable in a program running under the TSO terminal monitor (IKJEFT01) in batch, but there was no C@ENV DD statement defined. PERMANENT scope environment variables can be accessed in batch only when running under IKJEFT01, and only if a C@ENV DD statement is present.

Action

Supply the DD statement or correct the program not to attempt to access PERMANENT scope environment variables in batch.

---

725 **`WARNING: putenv failed due to improper sharing of TSO subpool 78. Errno value: EUSAGE`**

Explanation

A call to **`putenv`** or **`setenv`** failed because an ancestor task of the C program failed to share TSO subpool 78 properly with its subtask. Full SAS/C TSO environment variable support is dependent on proper sharing of subpool 78.

Action

Determine the program that is failing to share subpool 78 with its subtasks, and correct that program. If this is not possible, modify the program not to call **`putenv`** or **`setenv`** to update non-PROGRAM scope environment variables.

---

725 **`WARNING: putenv failed due to improper sharing of CICS temporary storage queue [name]. Errno value: EUSAGE`**

Explanation

Environment variables are stored in the first record of a temporary storage queue. The stored variables are preceded by an identifying header. In this particular instance, the temporary storage queue identified by [name] was found to contain a record which did not have a recognizable header. The SAS/C library assumes that the named queue is being used for some other purpose and cannot be used to store or retrieve environment variables.

Action

The default queue name used by the library is the VTAM luname associated with the transaction. If the named queue is already used for something other than storing environment variables, then modify the code in the user-replaceable module L$UEVQN to choose a different name for the library to use.

---

730 `WARNING: The [name] function is not supported in CICS.`
`Errno value: ESYS`

Explanation

A function was called that is not supported in CICS.

Action

Correct the program. The **iscics** function can be used to check whether or not the program is in a CICS environment. See the *SAS/C Library Reference, Volume 1*, for information on **iscics**.

---

800 `WARNING: Environment name [name] not acceptable to CLIST/Rexx.`
`Errno value: EARG`

Explanation

The environment name passed to execinit is not supported by the command language (CLIST or REXX under TSO, REXX under OpenEdition). The environment name is limited to 8 characters, which must be alphabetc, numeric or national ($, # or @) characters.

---

800 `WARNING/NOTE: CMS macro ''[name]'' failed with RC = [num]`
`Errno value: ESYS or unchanged if NOTE`

Explanation

The indicated CMS macro failed, producing the indicated return code. If the message is a NOTE, execution was able to proceed, ignoring the error. If the message is a WARNING, the function that detected the error returned failure to its caller.

Action

Refer to the *CMS Application Development Reference for Assembler* for an explanation of the return code and take appropriate action. If the return code indicates a programming problem, contact SAS/C Technical Support.

---

801 `WARNING: [function] rejected, a valid TSO or OpenEdition environment does not exist.`
`Errno value: EUSAGE`

Explanation

The **execinit** or **execmsg** function was called, but the program was not called with OpenEdition exec-linkage and was not running in a TSO address space with valid TSO control blocks.

Action

SUBCOM applications can only be run under TSO or when OpenEdition exec-linkage is in use (for instance, under the OpenEdition shell). To run a SUBCOM program in MVS batch, consider use of the background TMP (IKJEFT01).

---

801 `WARNING: Unknown parameter ''[name]'' in call to ''[routine]''`
`Errno value: ESYS`

Explanation

An invalid call was made to a REXX function package. The parameter list includes an argument that is not defined by REXX.

Action

Check that the function package has been called properly by REXX. If the call appears to be correct, contact SAS/C Technical Support.

---

802 `WARNING: NUCXLOAD command failed for ''[program]''`
`Errno value: ESYS`

Explanation

An error occurred when the CMS NUCXLOAD command was used by a REXX function package. This is probably a SAS/C internal error.

Action

Contact SAS/C Technical Support.

---

803 `WARNING: Attempt to call [name] without a previous successful call of execinit.`
`Errno value: EUSAGE`

Explanation

The program called a function which requires the prior use of **`execinit`** to define a SUBCOM environment. **`execinit`** was not called, or the call was unsuccessful.

Action

Correct the program to call **`execinit`** and to check the return code for failure.

---

804 `WARNING: Argument to execcall is not a valid EXEC command.`
`Errno value: EARG`

Explanation

The argument passed to **`execcall`** was not an implicit or explicit EXEC command. There may have been invalid characters in the command name.

Action

Correct the program to pass a valid command name.

---

805 `WARNING: execinit rejected, previous use of execinit is still active.`
`Errno value: EUSAGE`

Explanation

`execinit` was called twice in the same program without an intervening call to `execend`.

Action

Correct the program to call `execend` between calls to `execinit`.

---

806 `WARNING: Rexx input line longer than 32763 characters cannot be processed.`
`Errno value: ELIMIT`

Explanation

A very long line was read from a Rexx exec. Due to CLIST compatability requirements, the SAS/C SUBCOM interface cannot accept a line longer than 32763 characters.

Action

Rewrite the Rexx exec to use shorter lines.

---

806 `WARNING: Environment name is missing.`
`Errno value: EUSAGE`

Explanation

A call to `execinit` specified a null environment name. A valid environment name must be specified so it can be addressed in an EXEC.

Action

Correct the program. See the documentation on `execinit` in the *SAS/C Library Reference, volume 2* for more information on SUBCOM environment names.

---

807 `WARNING: Non-interactive execution inhibited.`
`Errno value: EUSAGE`

Explanation

The function `execinit` was called to initialize a non-interactive SUBCOM environment, but the program was not called directly from CMS. (For instance, it was called by an assembler language routine.) The non-interactive SUBCOM mode is supported only for programs called directly from CMS.

Action

Ensure that the application is called directly from CMS, removing any non-C front-ends. Alternately, redefine the application to use the interactive rather than the non-interactive SUBCOM mode.

---

808 **`WARNING: OpenEdition does not support non-interactive use of execinit.`**
`Errno value: EUNSUPP`

Explanation

The OpenEdition REXX implementation cannot support the non-interactive mode of the SAS/C SUBCOM support.

Action

Modify the application to specify interactive mode when calling **`execinit`**.

---

809 **`WARNING: Variable name: [name] or length: [num] is invalid.`**
`Errno value: EUSAGE`

Explanation

A variable name or length passed to the **`execshv`** function was invalid. The length is limited by CMS to 250 characters. The name is validated to make sure it contains only supported characters.

Action

Correct the program. See the *REXX/VM Reference* and the *CMS USer's Guide* publications for more information on valid EXEC variables.

---

810 **`WARNING: Unknown execshv function code: [num].`**
`Errno value: EARG`

Explanation

An invalid function code was passed to the **`execshv`** function.

Action

Correct the argument to **`execshv`** to use a symbolic name described in the documentation.

---

811 **`WARNING: This environment does not support the TSO CLIST variable access interface.`**
`Errno value: ENOSYS`

Explanation

The **`execshv`** function requires a version of TSO/E that supports the IKJCT441 interface routine.

Action

Run the program on a system with a recent version of TSO/E, or modify the program not to call the **`execshv`** function.

---

812 `WARNING: Length of Rexx/CLIST variable [name] is non-positive or too large: [num].`
`Errno value: EUSAGE`

Explanation

The arguments to **execshv** specified a negative variable name length or a length larger than supported by TSO.

812 `WARNING: The extended function names CSECT has been deleted.`
`Errno value: EUSAGE`

Explanation

While scanning the list of functions passed to **cmsrxfn**, the library found a function with an extended name. The extended names CSECT is not present, presumably because it was deleted during CLINK/COOL processing. **cmsrxfn** is therefore unable to determine the function names to be passed to REXX.

Action

Do not delete the extended names CSECT when linking a REXX function package. That is, specify the CLINK/COOL option **xfnmkeep** option, or allow it to default.

813 `WARNING: Invalid Rexx/CLIST variable name: [name]`
`Errno value: EUSAGE`

Explanation

A call to **execshv** specified an invalid variable name. The variable names are validated by TSO.

814 `WARNING: Update of CLIST variable not supported: [name]`
`Errno value: EUSAGE`

Explanation

A call to **execshv** requested that a CLIST variable be updated, but the CLIST language does not support update of this variable. Probably the variable is a CLIST **&SYS...** constant symbol such as **$SYSTIME**.

815 `WARNING: execget failed due to insufficent memory.`
`Errno value: ENOMEM`

Explanation

The **execget** function failed to return an input line to its caller because there was not enough memory to make a copy of the input.

Action

Ensure that sufficent memory is available.

816 **`WARNING: TSO Rexx/CLIST interface not supported in child process.`**
`Errno value: ESYS`

type 4 header

An attempt was made to use one of the functions **`execcall`**, **`execid`**, **`execget`** or **`execrc`** in a child process created by a TSO address space. A child process of a TSO address space is not itself a TSO address space, and therefore these functions can no longer be called.

Action

Change the program not to call any SUBCOM functions in a child process. The **`execend`** function can be called in this situation, and it will release any resources previously allocated by the TSO SUBCOM interface.

---

820 **`WARNING: Invalid DCB keyword: [keyword].`**
`Errno value: EARG`

Explanation

A call to **`osdcb`** specified an onvalid or unsupported DCB keyword.

Action

See *SAS/C Library Reference, Volume 2*, for a list of the valid keywords for **`osdcb`**.

---

821 **`WARNING: Data set [name] could not be opened, members not returned.`**
`Errno value: ESYS`

Explanation

`osdfind/osdnext` was unable to return the members for the named PDS because the PDS could not be opened.

Action

Check the job log for MVS messages giving further information about the failure. If running in TSO, specify PROFILE WTPMSG, which causes many MVS messages to be displayed at the terminal, and run again to get further information.

---

822 **`WARNING: Unexpected end of file on directory of PDS [name].`**
`Errno value: ECORRUPT`

Explanation

The directory of a PDS ended unexpectedly. This is probably an indication that the directory has been damaged.

Action

Report this message to your Systems Programming Staff, who may be able to recover the PDS or restore it from a backup copy.

---

823 `WARNING: I/O error reading directory of PDS [name].`
`Errno value: EDEVICE`

Explanation

An input error occurred reading the directory of the PDS listed. This may be a hardware error or may indicate software damage to the PDS.

Action

Report this message to your Systems Programming Staff, who may be able to determine the cause of the error and whether recovery is possible.

---

824 `WARNING: Invalid pattern for osdfind: [reason].`
`Errno value: EARG`

Explanation

The data set name pattern passed to **`osdfind`** was invalid. The text in the message is a more detailed explanation of the error, for instance, "name too long."

---

830 `NOTE: [name] subsystem unavailable for [name].`
`Errno value: unchanged`

Explanation

The VMCF subsystem of the IBM TCP/IP product is not running. Either IBM TCP/IP is not installed, or it has not been correctly initialized.

If your site runs a non-IBM TCP/IP product, this message indicates that your program has accessed the IBM TCP/IP interface module (LSCNCOM) rather than the module for your TCP/IP vendor's product.

Action

Consult your Systems Programming Staff. If your site runs a non-IBM TCP/IP product, ensure that your STEPLIB or CTRANS libraries are defined to access the correct version of LSCNCOM.

Also, you can use the **`setsockimp`** function to specify which TCP/IP implementation to use. See *SAS/C Library Reference, Volume 2* for more information on this function.

---

840 `WARNING: Unsupported dynamic allocation action code.`
`Errno value: EARG`

Explanation

The first argument to **`osdynalloc`** was not one of the defined dynamic allocation action codes.

Action

Correct the program to pass one of the symbolic action codes defined in **`<os.h>`**.

---

841 `WARNING: Unrecognized osdynalloc keyword: [name]`
`Errno value: EARG`

Explanation

The second argument to **`osdynalloc`** included a keyword not defined for the dynamic allocation action requested.

Action

Correct the argument.

---

842 `WARNING: osdynalloc keyword [name] must not specify a value.`
`Errno value: EARG`

Explanation

The **`osdynalloc`** keywords string included a keyword which cannot have an associated value, but a value was specified. For example, the string contained "tracks=5" rather than "tracks,space1=5".

Action

Correct the argument.

---

843 `WARNING: osdynalloc keyword [name] must specify a value.`
`Errno value: EARG`

Explanation

The **`osdynalloc`** keywords string included a keyword which requires an associated value, but no value was specified. For example, the string contained "spin" rather than "spin=unalloc".

Action

Correct the argument.

---

844 `WARNING: Value of osdynalloc keyword [name] is too long.`
`Errno value: EARG`

Explanation

The value specified by the named keyword in the **`osdynalloc`** keyword string was unreasonably long. Perhaps a comma has been unintentionally omitted.

Action

Correct the argument.

---

845 `WARNING: Value of osdynalloc keyword [name] is out of range. Errno value: EARG`

Explanation

The numeric value specified by the named keyword in the **osdynalloc** keyword string was outside the meaningful range for the keyword.

Action

Correct the argument.

---

846 `WARNING: Value of osdynalloc keyword [name] is not recognized. Errno value: EARG`

Explanation

The value specified by the named keyword in the **osdynalloc** keyword string was not recognized. The named keyword is restricted to a small set of values. For example the **DYN_ALLOC** keyword "status" is only allowed a value of "OLD", "NEW", "MOD" or "SHR".

Action

Correct the argument.

---

847 `WARNING: Character '[char]' in [name] keyword is not meaningful. Errno value: EARG`

Explanation

The value specified by the named keyword in the **osdynalloc** keyword string included an unrecognized character. The named keyword is restricted to a string of characters from a specific set. For instance, the specification "recfm=vby" is rejected because "y" is not a meaningful "recfm" specification.

Action

Correct the argument.

---

848 `WARNING: osdynalloc [name] value does not have the required format. Errno value: EARG`

Explanation

The value for the named keyword in the **osdynalloc** keyword string was not specified correctly. For example, the value for the "expdt" keyword did not match any of the permitted formats: yyddd, yyyyddd or yyyy/ddd.

Action

Correct the argument.

---

849 `WARNING: osdynalloc [name] keyword specified incorrectly.`
`Errno value: EARG`

Explanation

The value for the named keyword in the **`osdynalloc`** keyword string was not specified correctly. For example, the second item specified for the "secmodel" keyword was not "GENERIC".

Action

Correct the argument.

---

850 `WARNING: osdynalloc [name] keyword specifies a [invalid/negative] integer or pointer value.`
`Errno value: EARG`

Explanation

The value specified by the named keyword in the **`osdynalloc`** keyword string was not a valid positive number.

Action

Correct the argument.

---

851 `WARNING: No user id available, unable to complete [keyword] specification.`
`Errno value: EUSAGE`

Explanation

The value for the named keyword in the **`osdynalloc`** keyword string was specified as an incomplete data set name, starting with a period. The library was unable to determine the MVS user id associated with the job, and therefore the name could not be completed.

Action

Specify a complete data set name, or run the program in an environment where a user id can be determined.

---

852 `WARNING: Internal error in osdynalloc.`
`Errno value: ELIBERR`

Explanation

An unexpected situation was detected in **`osdynalloc`**. The allocation was not performed.

Action

Report this message to SAS/C Technical Support.

---

860 **`WARNING: [style] file name cannot be resolved, no userid can be determined.`**
`Errno value: ESYS`

Explanation

A request was made to open a **`''tso''`** or **`''cms''`** style file name in a non-TSO environment, and no userid could be determined. This usually means one of two things: the site is not running a RACF compatible security product; or the program is running as an MVS started task, and the security system has not been configured to define a userid for started tasks.

Action

Either run the program in an environment where a userid will be available, or use **`''dsn''`** or **`''ddn''`** style file names rather than **`''tso''`** names.

---

861 **`WARNING: A ''terminal'' file cannot be opened for input in batch.`**
`Errno value: EUSAGE`

Explanation

An attempt has been made to open the file name "*" for input in MVS batch. In batch, "*" is only supported for output (where it specifies the DDname SYSTERM).

Action

Modify the program to call the **`intractv`** function to determine whether it is running in batch or TSO, and use a different input file name in batch.

---

862 **`WARNING: Unable to allocate file. DYNALLOC macro return code [num] error reason code [hex] info [hex]`**
`Errno value: ESYS`

Explanation

While opening a **`''dsn''`** or **`''tso''`** style file name, dynamic allocation experienced an unexpected failure. System information associated with the failure is printed in the message.

Action

Look up the error codes in the *IBM MVS/ESA Application Development Guide: Authorized Assembler Language Programs* and take appropriate action. If the error codes indicate a programming problem, this is a SAS/C library internal error.

---

863 `WARNING: Existing file is not [organization].`
`Errno value: EUSAGE`

Explanation

The file to be opened did not have the required organization. This message most frequently occurs when the file name specifies a member name, but the data set is not a PDS.

Action

Specify the file name correctly.

---

864 `WARNING: File not created, [parm]=[value] conflicts with member name in file name.`
`Errno value: EUSAGE`

Explanation

The file to be opened does not exist, and an amparm was specified which cannot be used with a PDS, such as `''dir=0''`. However, the file name includes a member name. The file is not created.

Action

Correct the amparms or the file name.

---

865 `WARNING: Unable to deallocate file. DYNALLOC macro return code [num] error reason code [hex] info [hex]`
`Errno value: ESYS`

Explanation

During the opening of a `''dsn''` or `''tso''` style file name, the data set was allocated, and then open failed for some reason described by a previous message. When the library attempted to free the allocation after the previous error, the DYNALLOC SVC failed.

Action

Look up the error codes in the *IBM MVS/ESA Application Development Guide: Authorized Assembler Language Programs* and take appropriate action. If the error codes indicate a programming problem, this is a SAS/C library internal error.

---

866 **`WARNING: File not [action], [name] style prefix not supported by operating system.`**
`Errno value: EARG`

Explanation

Opening a file with a **`''dsn''`** or **`''tso''`** style file name requires the use of the DYNALLOC SVC, but this is not supported by the operating system.

Action

Do not use **`''dsn''`** or **`''tso''`** style file names under an operating system which does not support dynamic allocation.

---

867 **`WARNING: File not opened, [parm] amparm also requires [parms].`**
`Errno value: EARG`

Explanation

In a call to **`afopen`** or **`aopen`**, an amparm was used which requires another amparm to be meaningful. For instance, **`alcunit=block`** was specified, but the **`blksize`** amparm was not specified.

Action

Correct the program to specify all required amparms.

---

868 **`WARNING: Unable to allocate file. Request denied by installation validation routine.`**
`Errno value: ESYS`

Explanation

While opening a **`''dsn''`** or **`''tso''`** style file name, the library invoked the DYNALLOC SVC to allocate the file. This request was failed by the site's dynamic allocation exit.

Action

Contact your Systems Programming Staff to determine why this condition occurred and how to correct it.

---

869 **`WARNING: File name [name] cannot be [action].`**
`Errno value: EARG`

Explanation

An attempt was made to remove or rename a **`''dsn''`** or **`''tso''`** style file name which cannot be removed or renamed. Examples of such files include the terminal, NULLFILE, SYSOUT files and temporary data sets.

Action

Do not attempt to remove or rename files other than permanent disk files.

---

870 **`WARNING: Invalid [name] style file name.`**
`Errno value: EARG`

Explanation

A **`''tso''`** or **`''dsn''`** file name did not have the correct form.

871 **`WARNING: File not allocated, exclusive access required and not available.`**
`Errno value: EINUSE`

Explanation

An attempt was made to open a **`''tso''`** or **`''dsn''`** file name for writing. The file was already allocated and was shared with another job or user. The library was therefore unable to get exclusive use of the file.

Action

Run the program when no other users are using the file. Alternately, access the file via a DDname, which can be allocated SHR and shared with other users. (However, note that only one job or session should have the data set open at a time, or the effects are unpredictable.)

872 **`WARNING: File in use by another job or user.`**
`Errno value: EINUSE`

Explanation

An attempt was made to open a **`''tso''`** or **`''dsn''`** file name for writing. The library attempted to allocate the file OLD, but another job or user had the file allocted, so the attempt failed.

Action

Run the program when no other users are using the file. Alternately, access the file via a DDname, which can be allocated SHR and shared with other users. (However, note that only one job or session should have the data set open at a time, or the effects are unpredictable.)

873 **`WARNING: Volume [name] not mounted. Probable cause: volume invalid.`**
`Errno value: EARG`

Explanation

An attempt was made to open a **`''tso''`** or **`''dsn''`** file name, specifying the **`vol`** amparm. The volume specified was not or could not be mounted. This may indicate that the volume name was invalid.

Action

Correct the volume name, or contact Operations staff to determine why the volume was not mounted.

874 `WARNING: File not allocated, unit name [name] not defined.`
`Errno value: EARG`

Explanation

An attempt was made to open a `''tso''` or `''dsn''` file name, specifying the `unit` amparm. The unit specified needs to be defined.

875 `WARNING: File not allocated, [parmname] is invalid.`
`Errno value: EARG`

Explanation

One or more of the arguments to the failing function had incorrect syntax, for instance, the data set name, the member name, or the storage class.

Action

Correct the incorrect parameter.

876 `WARNING: File not allocated, [parmname] is too long.`
`Errno value: EARG`

Explanation

One or more of the arguments to the failing function was too long, for instance, the data set name, the member name, or the storage class.

Action

Correct the incorrect parameter.

877 `WARNING: File not allocated, request denied by operator.`
`Errno value: ESYS`

Explanation

Opening a `''dsn''` or `''tso''` style file name required operator action, but the operator cancelled the allocation request. This usually indicates that the allocation required mounting of an offline volume.

Action

Correct the file name not to require operator action, or consult with your Operations Staff on how to request operator action.

878 **`WARNING: File not allocated, request denied by operator. Probable cause: volume [name] invalid.`**
`Errno value: ESYS`

Explanation

Opening a **`''dsn''`** or **`''tso''`** style file name required operator action, but the operator cancelled the allocation request. This usually indicates that the volume requested using the vol amparm was not valid.

Action

Correct the volume name so that operator action is not required, or consult with your Operations Staff on how to request operator action.

---

879 **`WARNING: Data set does not exist: [name].`**
`Errno value: ENFOUND`

Explanation

The program requested opening a data set which does not exist for input. The data set name could not be found in either the system catalog or the list of currently allocated files.

Action

Correct the file name to specify an existing file, or open the file for writing as well as reading to allow the file to be created.

---

880 **`WARNING: File not allocated, disk space not available.`**
`Errno value: ENOSPC`

Explanation

The file requested could not be created due to insufficent disk space on suitable volumes.

Action

Lower the amount of space requested, or report this condition to your site's Systems Programing Staff for advice.

---

881 **`WARNING: File not allocated, dsname conflicts with the name of an existing file.`**
`Errno value: EUSAGE`

Explanation

An attempt was made to create a file whose name conflicts with the name of an existing file. For instance, an attempt was made to create FRED.MASTER.DATA when FRED.MASTER already exists. Note that this condition occurs only when the created file is cataloged in a CVOL.

Action

Consult with your Systems Programming Staff for information on how to circumvent this condition.

---

882 **`WARNING: File not allocated, conflicting specifications [parm]=[value] and [parm]=[value].`**
`Errno value: EUSAGE`

Explanation

Conflicting amparms were specified, for instance, **`''dir=5''`** or **`''keylen=8''`** and **`''org=ps''`**.

Action

Correct the conflicting amparms.

---

883 **`NOTE: Operating system does not support PDS/E. org=pdse treated as org=po.`**
`Errno value: EUSAGE`

Explanation

A request was made to create a data set as a PDS/E, but the operating system does not support PDS/E. (MVS/ESA 3.2 is required.) The data set is created as a regular PDS, and no error is indicated to the caller of **`afopen`**.

Action

None required. To avoid the message, correct the program not to request a PDS/E in an operating system which does not support them.

---

884 **`WARNING: A linear data set may not be opened for keyed access.`**
`Errno value: EARG`

Explanation

An attempt was made to open a linear data set (DIV object) for keyed access mode (open mode **`''rk''`**, **`''wk''`**, etc.) A linear data set can only be opened for binary access.

---

885 **`WARNING: Dynamic creation of VSAM files is not supported by this operating system.`**
`Errno value: ESYS`

Explanation

The library was unable to create a new VSAM data set because this level of the operating system does not support dynamic allocation of new VSAM files.

Action

Do not request dynamic creation of VSAM files unless your level of the operating system will support this.

---

886 **`WARNING: A non-zero key length must be specified when creating a VSAM KSDS.`**
`Errno value: EUSAGE`

Explanation

The library was unable to create a new VSAM data set because the amparms did not specify a key length or specified **`''keylen=0''`**. Note, if you open a non-existing file for keyed access, it is assumed to be a KSDS unless you use the **`org`** amparm to specify differently.

Action

Either specify a non-zero **`keylen`** value in the amparms, or modify the **`org`** amparm to request an ESDS or RRDS.

---

887 **`WARNING: The specification [amparm] is not supported by VSAM.`**
`Errno value: EARG`

Explanation

A VSAM-related amparm is outside the range of values accepted by VSAM. For instance, a key length greater than 255 was specified.

Action

Correct the amparm to be acceptable to VSAM. VSAM allows a maximum **`keylen`** of 255, a maximum **`keyoff`** of 32760, and a maximum **`reclen`** of 32760.

---

889 **`WARNING: File not opened due to library errors. Too many allocation text units.`**
`Errno value: ELIBERR`

Explanation

While attempting to open a file defined with a **`''dsn''`** or **`''tso''`** style name, the library suffered an internal error.

---

890 `WARNING: Unable to allocate new VSAM data set. Probable cause: SMS not active.`
`Errno value: ESYS`

Explanation

When the library attempted to create a new VSAM file, a sequential file was created instead. Most likely, SMS was not active.

Action

Report this error to your Systems Programming Staff.

---

891 `WARNING: A [name] style file name cannot be opened with ''org=[value].''`
`Errno value: EARG`

Explanation

An attempt was made to open a `''dsn''` or `''tso''` style file name specifying `''org=byte''`. This file organization is valid only for OpenEdition HFS files.

Action

Correct either the file name or the `org` amparm.

---

900 `WARNING: [function] failed, OpenEdition is not installed.`
`Errno value: EMVSNOTUP`

Explanation

OpenEdition must be installed and active to call this function. This message may also be generated if a program linked with a version of the SAS/C Library which did not support OpenEdition attempts to use an OpenEdition service, such as opening an HFS file.

Action

Do not use OpenEdition-dependent functions on a system where OpenEdition is not installed.

---

901 `WARNING: [function] failed, incorrect arguments passed by library, reason code [hex].`
`Errno value: ELIBERR`

Explanation

Erroneous arguments were passed to the OpenEdition interface for the function specified. This is probably a library internal error.

---

902 **`WARNING: [function] failed because [text], reason code [hex].`**
`Errno value: various`

Explanation

OpenEdition MVS detected a function failure. Possible reasons include "dynamic allocation error," "HFS permanent file error," and "SAF extract error." Check the *Assembler Callable Services for OpenEdition MVS* for further information. This message probably indicates a problem within OpenEdition MVS.

903 **`WARNING: Invalid argument to [function].`**
`Errno value: EINVAL or EFAULT`

Explanation

An invalid argument was passed to the named function.

Action

Check that all arguments have been specified properly. If the **`errno`** value was **`EFAULT`**, the invalid argument was a pointer which did not address a valid memory location.

904 **`WARNING: Unexpected failure in [function], reason code [hex].`**
`Errno value: various`

Explanation

When the library called OpenEdition to perform the specified function, an **`errno`** value not documented for that function was returned. This may be a runtime library error.

Action

Consult the list of **`errno`** values in Appendix 2, "ERRNO Values" and take appropriate action.

905 **`WARNING: Descriptor [num] is not associated with an open HFS file.`**
`Errno value: EBADF`

Explanation

The program called a function that requires an argument to be the file descriptor for an open HFS file.

906 **`WARNING: [function] failed, file was opened read-only.`**
`Errno value: EBADF`

Explanation

The indicated function requires a file descriptor that allows writing.

907 `WARNING: Descriptor [num] is not associated with a regular HFS file.`

`Errno value: EBADF`

Explanation

The program called a function that requires an argument to be a regular HFS file, that is, a disk file rather than a pipe, socket or terminal file.

Action

Correct the program to ensure that the argument to the failing function is the file descriptor for a regular HFS file.

---

908 `WARNING: File system [name] is already mounted.`
`Errno value: EINVAL`

Explanation

The program attempted to mount an already mounted file system.

Action

Do not mount a file system more than once.

---

909 `WARNING: Mount point [name] is the root of an already mounted file system.`
`Errno value: EINVAL`

Explanation

The first argument to the `mount` function specified a directory that was the target of a previous mount request.

Action

Do not attempt to mount more than one file system into the same directory.

---

910 `WARNING: The program owning the [name] file system has not been started.`
`Errno value: EINVAL`

Explanation

The call to the `mount` function specified a file system that was not activated when OpenEdition was initialized.

Action

Contact your Systems Programming Staff for assistance.

---

911 **`WARNING: [function] failed due to system limit on the number of open files.`**
`Errno value: ENFILE`

Explanation

A call to **`open`**, **`pipe`** or **`socket`** failed because a system limit of the total number of open HFS files was exceeded.

Action

Either modify your program to open fewer files and sockets, or contact your Systems Programming Staff to increase the system limit on open HFS files.

---

912 **`WARNING: Argument to [function] does not reference an open directory.`**
`Errno value: EBADF`

Explanation

An argument to **`readdir`**, **`rewinddir`**, or **`closedir`** does not correctly identify a directory stream opened with **`opendir`**. The argument may have been overlaid.

Action

Make sure the argument was passed to **`opendir`** before use and that it has not been inadvertantly modified.

---

913 **`WARNING: Overflow occurred during computation of times() results.`**
`Errno value: ERANGE`

Explanation

One or more of the values returned by **`times`** could not be returned accurately due to overflow. -1 was stored in place of the correct value. Overflow occurs if any value is greater than 2^{31} seconds.

---

914 **`WARNING: The file system to be unmounted is still in use.`**
`Errno value: EBUSY`

Explanation

A call to the **`umount`** function specified the argument **`MTM_UMOUNT`**, but the file system was still in use.

Action

If the file system must be unmounted despite being in use, specify another argument to **`umount`**. See the **`umount`** function description in *SAS/C Library Reference, Volume 2.*

---

915 `WARNING: Signal number not supported by OpenEdition: [num].`
`Errno value: EINVAL`

Explanation

The argument to the `kill` function is not the number of a signal supported by OpenEdition MVS.

Action

Correct the program to specify a signal supported by OpenEdition MVS. See "Signal Handling" in *SAS/C Library Reference, Volume 1.*

916 `WARNING: File descriptor [num] not valid for fcntl request [name].`
`Errno value: EBADF`

Explanation

The indicated file descriptor was not valid for the indicated `fcntl` request. For instance, to process a `F_SETLK` request for an exclusive lock, open the file descriptor for writing.

Action

Correct the `fcntl` call.

917 `WARNING: Attempt to [action] [action]-only file.`
`Errno value: EBADF`

Explanation

An attempt has been made to read a write-only file or to write a read-only file.

918 `WARNING: Ambiguous use of file descriptor [num].`
`Errno value: EBADF`

Explanation

An attempt was made to access a standard file descriptor (0, 1 or 2) that was allocated to some other file descriptor. The requested file descriptor number was not in use. This situation can arise only in unusual circumstances, generally involving the `freopen` function.

Action

Avoid referring to a file using a standard file descriptor number with which it is not normally associated.

919 `WARNING: Open failed, HFS pathname or pathname component too long.`
`Errno value: ENAMETOOLONG`

Explanation

A pathname passed to an OpenEdition function was too long or contained a component which was too long. The maximum length of a path name under OpenEdition is 1024 bytes; the maximum component length is 255.

920 `WARNING: Unable to open shadow file (/dev/null) for socket.`
`Errno value: various`

Explanation

When a non-integrated socket is accessed, the library opens a shadow file of **/dev/null** to prevent OpenEdition from reassigning the file number assigned to the socket. This diagnostic is issued if this open fails. Note that this condition also causes the socket access to fail.

Action

Consult the list of **errno** values in Appendix 2, "ERRNO Values" and take appropriate action.

---

921 **WARNING: Descriptor [num] is not an open HFS file or socket.**
Errno value: EBADF

Explanation

The first argument to **fcntl** must be an open HFS file descriptor or an open socket descriptor.

---

922 **WARNING: Invalid file descriptor: [num].**
Errno value: EBADF

Explanation

This message probably indicates an internal library error.

---

923 **WARNING: fdopen failed, open mode incompatible with the file descriptor.**
Errno value: EFATTR

Explanation

fdopen either was set to read in a file descriptor that was not opened for reading or set to write in a file descriptor that did not permit writing.

---

924 **NOTE: Access method parameter [parm] ignored for HFS file.**
Errno value: unchanged

Explanation

The amparm indicated was not meaningful for an HFS file and was ignored. However, the file will be opened successfully if no more serious conditions are detected.

---

925 `WARNING: DD statement specifies unsupported file type: [ddname].`
`Errno value: EFATTR`

Explanation

An attempt was made to open a DDname, but the DDname specified an unusual file type which could not be opened.

Action

Correct the DD statement to specify a valid MVS or HFS file.

---

926 `WARNING: DYNALLOC macro error: return code [num], error reason code [hex] info [hex].`
`Errno value: ESYS`

Explanation

The library issued DYNALLOC to obtain information about a DD statement allocated to an HFS file, but DYNALLOC failed as indicated.

Action

Look up the error codes in the *IBM MVS/ESA Application Development Guide: Authorized Assembler Language Programs* and take appropriate action. If the error codes indicate a programming problem, this is a SAS/C library internal error.

---

927 `WARNING: File cannot be opened, DD statement specifies [action] only.`
`Errno value: EFATTR`

Explanation

An attempt to open a DDname allocated to an HFS file failed because the open mode conflicted with the PATHOPTS specified on the DD statement. For example, the program attempted to open the file for reading, but PATHOPTS specified OWRONLY.

Action

Correct the program or the DD statement as appropriate.

---

928 `WARNING: File not allocated, environment variable [name] specifies an invalid data set name.`
`Errno value: EFATTR`

Explanation

An **exec**-linkage program attempted to open a DDname defined by an environment variable, but the value of the environment variable is not a valid MVS data set name.

Action

Correct the environment variable to specify a valid data set name or HFS file name.

---

929 **`WARNING: DD name [ddname] not allocated - [reason].`**
`Errno value: various`

Explanation

An **`exec`**-linkage program attempted to open a DDname identified by an environment variable, but the file could not be allocated. Possible reasons for this failure include "data set name invalid," "member name invalid," "data set not found," "data set in use," "reserved DD name," and "denied by installation exit."

Action

If the reason is "data set name invalid," "member name invalid," or "data set not found," correct the environment variable to specify a valid existing MVS data set name. If the reason is "reserved DD name," use another DDname. If the reason is "data set in use," assure that the data set is not in use by another job or session before running the program. If the reason is "denied by installation exit," consult your Systems Programming Staff.

---

930 **`WARNING: [ddname] allocation failure - DYNALLOC return code [num], error code [hex] info [hex].`**
`Errno value: ESYS or ELIBERR`

Explanation

An **`exec`**-linkage program attempted to open a DDname identified by an environment variable, but the file could not be allocated due to an unusual condition. If the **`errno`** is **`ELIBERR`**, the failure was an internal library error.

Action

Look up the error codes in the *IBM MVS/ESA Application Development Guide: Authorized Assembler Language Programs* and take appropriate action. If the error codes indicate a programming problem, this is a SAS/C library internal error.

---

931 **`WARNING: File [pathname] not allocated to DD name [ddname] - OpenEdition reason code [hex].`**
`Errno value: various`

Explanation

An **`exec`**-linkage program attempted to open a DDname identified by an environment specifying an HFS file, but the file could not be allocated. The **`errno`** value and reason code identify the cause of the failure.

Action

Consult the list of **`errno`** values in Appendix 2, "ERRNO Values" and take appropriate action. The reason code can be looked up in *Assembler Callable Services for OpenEdition MVS* for further information. This message may indicate a problem within OpenEdition MVS.

---

932 `WARNING: [function] failed due to insufficent memory.`
`Errno value: ENOMEM`

Explanation

The **fork** or **oeattach** function failed due to insufficient memory.

Action

Report this error to Systems Programming Staff. Note that it is unlikely that this condition can be corrected by specifying a larger region size.

933 `WARNING: [function] failed due to insufficent resources or environmental error - reason code [hex].`
`Errno value: EAGAIN`

Explanation

An error occurred in the **fork** or **oeattach** function. This message may indicate a temporary condition.

Action

Look up the reason code in *Assembler Callable Services for OpenEdition MVS* for further information. This message may indicate a problem within OpenEdition MVS.

934 `WARNING: fork not permitted - conflicts with =multitask runtime option.`
`Errno value: EFORBID`

Explanation

A program running with the **=multitask** runtime option invoked the **fork** function, which is not compatible with this option.

Action

If the program requires the use of **fork**, run it without the **=multitask** option.

935 `WARNING: File is no longer accessible in child process.`
`Errno value: ESYS`

Explanation

The child of a SAS/C program attempted to access an MVS file opened by its parent. Only HFS files opened in the parent can be accessed in a child.

Action

Correct the child so it does not access MVS files opened by the parent, other than to close them.

936 **`WARNING: Function not supported for [style] style file names.`**
`Errno value: EINVAL`

Explanation

A function was called for a file whose name specified an invalid type of file, for instance, a style which does not support directories for the **`chdir`** function.

Action

Correct the program so it specifies an appropriate file name. Note that if the application is not compiled with the **`posix`** compiler option, an explicit style prefix of **`''hfs:''`** is required to access an HFS file (unless **`''hfs''`** is the default style).

---

937 **`WARNING: Socket is no longer accessible in child process.`**
`Errno value: ESYS`

Explanation

The child of a SAS/C program attempted to access a non-integrated socket opened by its parent. Only integrated sockets opened in the parent can be accessed in a child.

Action

Avoid accessing a non-integrated socket opened by a parent in a child process except to close it. Consider the use of integrated sockets for programs which call **`fork`**. See *SAS/C Library Reference, Volume 2* for information on the **`setsockimp`** function, which can be used to request the use of integrated sockets.

---

938 **`WARNING: Improper file mode for popen: [mode].`**
`Errno value: EINVAL`

Explanation

A call to **`popen`** specified a file mode that permitted both reading and writing. Only a read-only or a write-only file mode may be specified.

Action

Correct the call to **`popen`**.

---

939 **`WARNING: The argument to pclose is not a file opened by popen.`**
`Errno value: EBADF`

Explanation

A call to **`pclose`** specified a **`FILE`** pointer which was not opened by the **`popen`** function.

Action

Use the **`fclose`** function to close **`FILE`** pointers not opened by **`popen`**.

---

940 `NOTE: pclose incomplete due to termination by longjmp.`
`Errno value: unchanged`

Explanation

While **pclose** was waiting for a child process to terminate, a signal was received; the signal handler called **longjmp**, terminating **pclose** without allowing the child process to terminate.

Action

The program continues to execute, but execution may be affected because the child process continues to run.

---

941 `NOTE: system function terminated by longjmp - child process remains active.`
`Errno value: unchanged`

Explanation

While the **system** function (called to invoke an OpenEdition shell command) was waiting for a child process to terminate, a signal was received; the signal handler called **longjmp**, terminating **system** without allowing the child process to terminate.

Action

The program continues to execute, but execution may be affected because the child process continues to run.

---

942 `WARNING: This function is not supported by your version of OpenEdition.`
`Errno value: ENOSYS`

Explanation

A function was called which requires a later version of OpenEdition than the version installed at your site.

Action

Note that the **envlevel** function can be used to determine the level of OpenEdition, for programs called with **exec**-linkage.

---

943 `WARNING: OpenEdition returned undocumented return code.`
`Errno value: various`

Explanation

An OpenEdition function returned an **errno** value whose meaning was not defined at the time this book was written. This message is only generated if the **=warning** runtime option is specified.

Action

Consult the latest version of the *MVS/ESA Application Development Reference: Assembler Callable Services for OpenEdition MVS* to determine the meaning of the **errno** value.

---

944 `WARNING: OpenEdition failure: [reason].`
`Errno value: various`

Explanation

An error in the OpenEdition TCP/IP interface occurred. This may be an error in OpenEdition or a hardware problem.

Action

Consult the latest version of the *MVS/ESA Application Development Reference: Assembler Callable Services for OpenEdition MVS* to determine the meaning of the `errno` value.

---

945 `WARNING: Descriptior [num] is not associated with an OpenEdition socket.`
`Errno value: ENOTSOCK`

Explanation

A socket interface function was called with an invalid file descriptor or one which identified an open HFS or MVS file rather than a socket.

---

946 `WARNING: Specified addressing family not supported.`
`Errno value: EAFNOSUPPORT`

Explanation

A socket interface function was called that was not supported for the socket's addressing family.

---

947 `WARNING: [function] failed, unable to allocate socket buffer space.`
`Errno value: ENOBUFS`

Explanation

A socket interface function was unable to obtain memory for buffers.

Action

Report this condition to your Systems Programming Staff.

---

948 `WARNING: No call to listen prior to accept.`
`Errno value: EINVAL`

Explanation

A call to **accept** was made without a previous call to **listen** for the same socket.

Action

Correct the program to call **listen** before calling **accept**.

---

949 `WARNING: connect failed, socket already connected.`
`Errno value: EISCONN`

Explanation

A call to **connect** was issued for an already connected socket.

---

950 `WARNING: accept expected due to prior listen.`
`Errno value: EOPNOTSUPP`

Explanation

A call to **connect** was issued for a socket previously passed to **listen**.

Action

Correct the program to call **accept** after calling **listen**.

---

951 `WARNING: [function] failed, argument not a stream socket.`
`Errno value: EPROTOTYPE`

Explanation

The indicated function is only meaningful for a stream socket.

---

952 `WARNING: [function] failed, socket not connected.`
`Errno value: ENOTCONN`

Explanation

The indicated function is meaningful only for a connected socket.

---

953 `WARNING: [function] failed, AF_UNIX not supported.`
`Errno value: ENOSYS`

Explanation

The indicated function is not supported for **AF_UNIX** sockets.

Action

Do not call **getsockopt** or **setsockopt** for **AF_UNIX** sockets.

---

954 `WARNING: listen failed, socket not bound or already listening.`
`Errno value: EINVAL`

Explanation

The **`listen`** function was called for a socket which had not already been bound (by a call to **`bind`**), or for which **`listen`** had already been called.

955 `WARNING: [function] failed, socket has been shutdown for input.`
`Errno value: EINVAL`

Explanation

An attempt has been made to receive data on a socket which has been shutdown for input.

956 `WARNING: [function] failed, message too large to send as a datagram.`

`Errno value: EMSGSIZE`

Explanation

An attempt was made to send a message on a datagram socket that was too large.

Action

Decrease the size of the message, or consult your Systems Programming Staff about increasing the limit.

957 `WARNING: [function] failed, incorrect socket protocol.`
`Errno value: EPROTOTYPE`

Explanation

A socket write request failed due to use of an incompatible protocol.

Action

Correct the program to use a compatible protocol.

958 `WARNING: Permission to create socket denied.`
`Errno value: EACCES`

Explanation

Permission was denied to create a socket of the specified type and protocol.

Action

Inform your Systems Programming Staff of this condition.

959 `WARNING: Specified protocol not supported for domain or socket type.`

`Errno value: EPROTONOSUPPORT`

Explanation

The protocol requested in a call to **`socket`** is not supported for the specified domain and socket type.

---

960 `WARNING: Unexpected subtask failure during oeattach processing.`
`Errno value: ESYS`

Explanation

A library subtask used by the **`oeattach`** function terminated unexpectedly.

Action

Contact SAS/C Technical support. Check the console log for any unusual messages which may provide further information about the nature of the failure.

---

961 `WARNING: OpenEdition services not available to this task.`
`Errno value: EMVSINITIAL`

Explanation

The library determined that it was not possible for the program to access any OpenEdition services. The most likely cause of this message is that your site has not defined your RACF user id or RACF group id to have an OpenEdition uid or gid.

Action

Contact your system administrator to request authorization to use OpenEdition services. If your user id and group id are properly defined for OpenEdition, contact SAS/C Technical Support.

---

963 `WARNING: This function is not enabled at this site.`
`Errno value: ENOSYS`

Explanation

The function **`chpriority`** or **`setpriority`** was called, but the site has not enabled the OpenEdition support for these functions.

Action

Inform your Systems Programming Staff if you require enablement of the OpenEdition support for these functions.

---

964 **`WARNING: The argument to <function> is not a regular HFS file.`**
`Errno value: EINVAL`

Explanation

The program called a function that requires an argument to be a regular HFS file, that is, a disk file rather than a pipe, socket, or terminal file.

Action

Correct the program to ensure that the argument to the failing function is the name of a regular HFS file.

---

980 **`WARNING: oesigsetup must be called before any other signal-handling function.`**
`Errno value: EUSAGE`

Explanation

A call to **`oesigsetup`** occurred after some other signal-handling function was called. Except for signal set manipulation functions like **`sigemptyset`** and **`sigaddset`**, **`oesigsetup`** must be the first signal-handling function call. The late call to **`oesigsetup`** is ignored, and the default division of signals between the SAS/C library and OpenEdition MVS is in effect.

Action

Move the **`oesigsetup`** call so it precedes any other signal-related calls.

---

981 **`WARNING: OpenEdition and native signal sets overlap.`**
`Errno value: EINVAL`

Explanation

In a call to **`oesigsetup`**, one or more signals are specified in both the SAS/C signal set and the OpenEdition signal set. The call to **`oesigsetup`** has no effect.

---

982 **`NOTE: Function [function] not supported when SIGALRM managed by OpenEdition.`**
`Errno value: unchanged`

Explanation

A call was made to the **`alarmd`** or **`sleepd`** function in a program for which **`SIGALRM`** is managed by OpenEdition. These functions are supported only when SIGALRM is managed by SAS/C. The time interval is rounded down to the nearest second, and the corresponding standard function is called; in the case of **`alarmd`** with a non-zero argument less than 1, the signal **`SIGLARM`** is raised immediately.

Action

Either change the program to allow management of **`SIGLARM`** by SAS/C, or replace the call to **`alarmd`** or **`sleepd`** with a call to **`alarm`** or **`sleep`**. Also note that the **`select`** function can often be used in place of **`sleepd`**.

---

983 `NOTE: MVS SIGINT handling is not supported in OpenEdition address space.`
`Errno value: unchanged`

Explanation

An attempt has been made to define a handler for the **SIGINT** signal. **SIGINT** is defined as managed by SAS/C, but the program was invoked by **exec** or is a child process created by **fork**. SAS/C **SIGINT** handling is effective only in a TSO address space.

Action

Either define **SIGINT** as managed by OpenEdition, or avoid defining a **SIGINT** handler in a non-TSO address space. The **intractv** function can be used to test whether a program is running in a TSO address space. See *SAS/C Library Reference, Volume 2.*

984 `WARNING: [name] signal cannot be handled - no oesigsetup call.`
`Errno value: EFORBID`

Explanation

A **sigaction** or **signal** call has been made for an OpenEdition signal, but the program does not have **exec**-linkage, and no call to **oesigsetup** has been made. By default, a non-**exec**-linkage program is not able to handle OpenEdition signals.

Action

Call the **oesigsetup** function to permit signals to be managed by OpenEdition before calling any other signal-handling function. See "Signal Handling" in *SAS/C Library Reference, Volume 1.*

990 `WARNING: Rexx execution terminated by [name] signal.`
`Errno value: EINTR`

Explanation

The process running the SAS/C REXX interface for OpenEdition was terminated by a signal. Note that when this message is issued, the interface to OpenEdition REXX has been disabled and any call other than **execend** will fail.

Action

Determine the cause of the signal. If the signal was not intentional, contact SAS/C Technical Support.

991 **`WARNING: Rexx execution terminated by [name] signal, ABEND code [code].`**
`Errno value: EINTR`

Explanation

The process running the SAS/C REXX interface for OpenEdition was terminated by an ABEND. Note that when this message is issued, the interface to OpenEdition REXX has been disabled and any call other than **`execend`** will fail.

Action

Contact SAS/C Technical support.

---

992 **`WARNING: Rexx environment no longer active due to previous error.`**
`Errno value: EPREV`

Explanation

A previous call to the SAS/C SUBCOM interface failed, generating message 990 or 991 and **`errno EINTR`**. No further calls to the SUBCOM interface (other than **`execend`**) can be made.

Action

Correct the program to check for **`errno EINTR`** after the failure of a SUBCOM function.

---

993 **`WARNING: OpenEdition Rexx interface failed due to failure of attach_execmvs service.`**
`Errno value: various`

Explanation

The **`execinit`** function could not complete because OpenEdition **`attach_execmvs`** service failed. A message may have been generated describing the failure of the service.

Action

If there were other messages describing the failure, take the appropriate action for these messages. If there were no other messages, contact SAS/C Technical Support.

10 OMD Messages

The OMD370 utility may produce messages for either invalid object decks created by the compiler, or problems related to input parameters, which should be corrected by the site programmer. Persistent problems with the OMD370 utility or invalid object decks should be reported to SAS/C Technical Support. The diagnostics described in this chapter typically can be corrected by the user.

The general format of an OMD message is:

```
*** Warning: [message text]
       or
*** Error: [message text]
```

Unnumbered Messages

Can't create OMD listing.

Explanation

An error occurred when OMD370 tried to open the listing file. This can be caused, for instance, by specifying a write-protected location for the file. Except on UNIX, this message is usually accompanied by a message from the run-time library, which contains more specific information about the problem. Correct the problem and re-run OMD370.

Can't open [object | source] file [name | none specified].

Explanation

OMD370 could not open the specified file. In UNIX, if no name is given on the command line, the name shown will be "none specified." Ensure that the file exists and re-run OMD370.

Invalid DDname prefix specified: prefix

Explanation

MVS only: The prefix specified in the FILES option was not a correct prefix for a DDname. The prefix must be between 1 and 3 characters. The first character of the prefix must be alphanumeric, or @, #, or $. The remaining characters must be alphanumeric.

Invalid OMD option: option

Explanation

OMD370 did not recognize the specified command line token as an option. Refer to the documentation for OMD370 for a list of OMD370 options.

`Invalid pagesize: [num]. Set to 55.`

Explanation

The value specified by the PAGESIZE (-p) option was less than 10 or greater than 255. OMD370 resets the value to the default, 55 lines per page. Specify a value in the accepted range.

`No line number/offset table.`

Explanation

The compiler's **`nolineno`** option was in effect when the object module was generated. This causes the compiler to suppress the generation of the line number/offset table CSECT that OMD370 uses to match source line numbers to offsets in the generated code. Recompile the program using the **`lineno`** option. **`lineno`** is the default.

`Not enough memory to continue.`

Explanation

An OMD370 storage allocation could not be satisfied. If you are running in MVS batch, increase the region size for the job. If you are running in TSO, increase the TSO region size. If CMS, increase the size of your virtual machine.

`Premature end of source file.`

Explanation

The line number/offset table CSECT in the object file specified a line number that exceeds the number of lines in the source file. This can occur when the specified object file is not associated with the specified source file. This can also occur if a SAS/C source file that has been generated by the C++ translator is compiled but the CXX option is not used. Ensure that the source file and object file match or, if the source file was generated by the C++ translator, the CXX option is used when the source file is compiled.

`TEXT filename missing or invalid.`

Explanation

CMS only: The filename specified as the first non-option argument to OMD370 is missing or cannot be interpreted as a CMS filename. Check that the filename is not preceded with a - (dash) and contains no invalid characters.

`TXTLIB filename not specified with LIB option.`

Explanation

CMS only: The LIB (-l) option was used but no filename was found. Re-run OMD370, specifying the name of a text library (having the filetype TXTLIB) following the LIB (-l) option.

`Unknown card type in object file.`

Explanation

OMD370 detected a record in the input file that is not a valid object record type. Check to make sure that the input file is an object file generated by the compiler or a 370 assembler such as Assembler H or HLASM.

11 DSECT2C Messages

All error conditions cause termination of execution.

Unnumbered Messages

`ERROR: No DSECT name has been specified on the command line.`

Explanation

Specify the name of the DSECT to be converted as part of the DSECT2C command line or PARMS field.

`ERROR: Memory is not available to evaluate expression.`

Explanation

DSECT2C ran out of memory while trying to evaluate an expression. This is probably caused by an ill-formed expression.

`ERROR: Invalid arithmetic operation has been specified.`

Explanation

Only those operations accepted by the assembler can be used in expressions. This message does not occur unless the assembler has also issued a diagnostic message for the expression.

`ERROR: Unbalanced parentheses.`

Explanation

Only those expressions accepted by the assembler can be used. This message does not occur unless the assembler has also issued a diagnostic message for the expression.

`ERROR: Invalid expression type.`

Explanation

Only those expressions accepted by the assembler can be used. This message does not occur unless the assembler has also issued a diagnostic message for the expression.

`ERROR: Invalid operand type.`

Explanation

Only those expressions accepted by the assembler can be used. This message does not occur unless the assembler has also issued a diagnostic message for the expression.

`WARNING: Undefined symbol used in expression.`

Explanation

A symbol used in an expression was not defined in the DSECT to be converted or in an EQU instruction preceding the DSECT. DSECT2C will not inspect other DSECTs in the input file for undefined symbols.

`ERROR: Out of memory. Program must halt.`

Explanation

DSECT2C keeps a copy of the DSECT in memory, as well as the data required to generate the structure. Very large DSECTs may require you to run DSECT2C in a larger virtual machine (under CMS) or region (under MVS).

`ERROR: DSECT name not found in listing.`

Explanation

The DSECT name specified on the command line cannot be found in the input file. This is probably caused by a spelling error.

`ERROR: Invalid file name used for input file.`

Explanation

The input file could not be opened. Check **`stderr`** for library warning messages containing more information.

`ERROR: Invalid file name used for output file.`

Explanation

The output file could not be opened. Check **`stderr`** for library warning messages containing more information.

`ERROR: Error writing to stdout. See stderr for more information`.

Explanation

An error occurred while writing the converted DSECT to the file associated with **`stdout`**. In MVS/TSO and CMS, the library will have written a warning message containing more information about the error.

WARNING: Undefined parameter detected.

Explanation

A token in the command line, other than the first, does not start with a hyphen (-) or does not correspond to a DSECT2C option.

WARNING: Symbol longer than 63 characters - will be truncated.

Explanation

A symbol in the input DSECT is longer than DSECT2C's limit of 63 characters. DSECT2C truncates the symbol on the right when it converts the symbol to a C indentifier.

12 LSCD Debugger Messages

This chapter documents messages generated by the SAS/C Debugger. Each message contains the message text, an explanation of the message, system information, and the action needed to correct the error, if applicable.

Debugger diagnostic messages have the following form:

```
LSCD[num] [message text]
```

For invalid syntax messages, see *SAS/C Debugger User's Guide and Reference*, which contains descriptions of the commands that are supported by the debugger. A brief description of the commands is also provided by the SAS/C Debugger help system.

LSCD Messages

000 **`An invalid command name was specified. (Enter ? for a list of valid command names).`**

Explanation

A command was entered but was not a valid debugger command.

002 **`The argument specified for the ON/BREAK/TRACE/IGNORE/QUERY command is not valid.`**

Explanation

An invalid argument was specified.

004 **`ON expects a single command or a command list within braces.`**

Explanation

An `on` command was issued with more than one deferred command without braces.

005 **`Section-name and ) should follow (.`**

Explanation

A left parenthesis was found, but the next token was not a name or an opening parenthesis was found that was not followed by a closing parenthesis.

007 `The function-name or section-name argument should be ''calls'', ''entry'', ''return'', ''*'' or a line-number range.`

Explanation

An invalid **on** command was entered. The syntax begins
on *name* [**calls** | **entry** | **entry** | * | *line-number*]

008 `There is invalid text, ''[text]'', after the end of the when clause.`

Explanation

A **when** clause was entered with text after the *expression* argument.

009 `The [command] command without arguments is only valid in an ON command list. The command is ignored.`

Explanation

A [command] command was entered without any arguments. This format is supported only as an operand to the **on** command where [command] values include **break** and **trace**.

010 `The expression specified is incomplete.`

Explanation

A **whatis** command was issued, but the expression specified was not complete.

011 `''[keyword]'' is not a valid keyword for the [command] command.`

Explanation

The following table lists the valid keywords for each of the commands that accept a keyword argument.

[command]	[keyword]
auto	cmacros, nocmacros, cxx, nocxx, dumpabs,nodumpabs, echo, noecho, exececho, noexececho, extname, noextname, id, noid, japan, nojapan, linesize, list, nolist, nullptr, nonullptr, wrap, nowrap
config	file, save
log	append, capture, file, start, stop
set	search, cache

013 `[name-type] name has been truncated to [len] characters.`

Explanation

An **install**, **define**, or **undefine** command was issued that specified either a user command, an EXEC or CLIST, or a macro with a name that was too long. User commands have a limit of 40 characters and EXECs, CLISTs, macros have a limit of 8 characters.

014 `End of range ([eor]) cannot be less than start ([sor]).`

Explanation

A **drop**, **enable**, or **disable** command was entered, but the operands specified, [start-of-range]:[end-of-range] with [end-of-range]<[start-of-range], are not supported.

015 `Invalid TYPE/cast specified.`

Explanation

While parsing a cast expression, invalid syntax was found.

016 `An integer must follow the COUNT keyword.`

Explanation

An **on** command was issued with the **count** clause, but **count** was not followed by an integer.

017 `The syntax specified for the [command] command is not valid.`

Explanation

The [command] command was issued, but was not valid. Additional messages are generally issued that further explain the problem.

[command] values:
install, **drop**, **enable**, **disable**, **monitor**, **print**, **auto**, **config**, **define**, **exec**, percent (**%**), **help**, **undefine**, **system**, **browse**, **log**

018 `[command] has not been installed as a user command. It already exists.`

Explanation

An **install** [command] command was issued, but [command] already exists.

019 `End-of-input occurred before continuation was completed.`

Explanation

While reading a line a continuation character was encountered, but no continuation input was found.

020 **`Logical input line cannot exceed 312 characters - input discarded.`**

Explanation

A line longer than 312 chars was read from either a CLIST, EXEC, stack, or user input. The input is discarded.

---

021 **`An invalid TYPE/cast was specified: a right parenthesis is missing.`**

Explanation

While parsing a **`type`** or **`cast`** expression, invalid syntax was found. Either a right parenthesis or a colon was missing.

---

022 **`A length of 0 may not be specified as the number of bytes to be DUMPed.`**

Explanation

A **`dump`** command was issued that specified a length of zero.

---

023 **`There is an invalid symbol in the specified list.`**

Explanation

Operands in a list must be separated by commas, but a comma was not found where expected.

---

024 **`Only a non-zero number of bytes may be specified.`**

Explanation

A **`copy`** command was issued that either did not specify the number of bytes, or specified zero bytes to copy.

---

025 **`The when clause specified is not valid.`**

Explanation

An **`on`** command was issued that specified an invalid **`when`** clause.

---

026 **`There is invalid text, ``[text]'', after the end of a valid expression.`**

Explanation

A **`when`** expression was specified that had extra text.

---

027 `Nested braces are not allowed in a list.`

Explanation

While parsing a list of items, an apparent nested brace was found. This is not supported.

029 `Only plain types and enum types can be specified with a value list.`

Explanation

A **COPY** command was entered, but a **plain** type or **enum** type was not specified with a value list.

030 `Missing function-name, section-name, or hook / event specification.`

Explanation

A required name or specification is missing.

031 `Continue input line ...`

Explanation

While reading a line, a continuation symbol was encountered before the end of line.

Action

Continue entering the input line.

032 `unsigned/signed float is an invalid type.`

Explanation

While parsing a cast expression, an **unsigned** or **signed float** type was found.

033 `unsigned/signed double is an invalid type.`

Explanation

While parsing a cast expression, an **unsigned** or **signed double** value was found.

034 `The following input in the buffer has been discarded: [input]`

Explanation

The debugger decided to flush the input line as shown in the next message.

035 `An invalid command name was specified.`

Explanation

A command was entered, but it was not a valid debugger command.

038 **`A range may not be specified in full-screen mode.`**

Explanation

A **`list`** command was issued that specified a line number range, but only a starting line is allowed in full-screen mode.

039 **`The [command] command without parameters is invalid.`**

Explanation

The **`auto`**, **`config`**, **`window`**, or **`log`** command was issued without parameters.

Action

Specify a valid parameter.

041 **`A section-name must follow a '('.`**

Explanation

A **`list`** command was entered with an open parenthesis. This indicates the specification of a *section-name*, but a *section-name* was not found.

042 **`A ')' must follow a section-name.`**

Explanation

A **`list`** command was issued with an open parenthesis and a section-name, but a closing parenthesis was not found.

043 **`An integer or '*' must follow a function-name or a parenthesized section-name.`**

Explanation

A **`list`** *function-name* | (*section-name*) command was issued, but it was not followed by a line number specification.

044 **`A ':' must follow an integer if a line-number range is to be specified.`**

Explanation

A **`list`** command was issued and was followed by another number. If a line number range was desired, it must be specified with a colon between the starting line number and the ending line number. Line number ranges are used only in line mode.

045 **`An integer or a signed positive integer must follow a ':'`**

Explanation

A **`list`** command was issued that specified a starting line number followed by a colon. The debugger expects a line number range, and an ending line number must be specified after the colon. Line number ranges are used only in line mode.

046 `An integer must follow a '+' or a '-'`

Explanation

A **list** command was specified with starting line number followed by a colon and then either a plus sign or a minus sign. An integer must follow the sign specification to indicate a range of lines. This form of the **list** command is only used in line mode.

047 `A signed positive integer must follow a ':'`

Explanation

A **list** command was issued with a line range specification that gave a positive start-of-range and a negative end-of-range. Since the start-of-range is positive, the end-of-range must be positive also.

048 `A signed integer must follow a ':'`

Explanation

A **list** command was issued with a line range specification that gave a negative start-of-range, but it did not use a signed integer for the end-of-range. Since the start-of-range was specified as a signed integer, the end-of-range must also be a signed integer.

049 `The first argument of the LIST command is invalid.`

Explanation

A **list** command was entered with invalid arguments.

050 `A line number of 0 is invalid.`

Explanation

A **list 0** command was issued, but the first line in a file is 1, not zero.

051 `The [command] command is not supported under OpenEdition.`

Explanation

The [command] command was entered under OpenEdition, but this command is not supported. Commands not supported include: percent (**%**), **config**, **dbinput**, **dblog**, **keys**, **transfer**, **watch**, and **window**.

052 `Record read from DBGIN longer than 256.`

Explanation

Logical Record length of `DBGIN` is greater than 256.

Action

Ensure that the file that is associated with DBGIN has a logical record length of less than or equal to 256.

---

053 `Identifier truncated to [size] characters.`

Explanation

While parsing an expression, a token was found with more than [size] characters. The specified identifier was truncated.

---

054 `The following input has been discarded: [line]`

Explanation

The debugger decided to flush the input line [line].

---

055 `Cannot PRINT ''[var]'' as the dimension of the array is not available.`

Explanation

A `print` command was entered with the array variable [var], but the dimension was not specified.

---

056 `A comma is missing or there are too few function arguments.`

Explanation

For a function specified in an expression, either a comma is missing or there are too few arguments.

---

057 `A right parenthesis is missing or there are too many function arguments.`

Explanation

For a function specified in an expression, either a right parenthesis is missing or there are too many arguments.

Action

See the definition of the function for the proper argument list.

---

059 **`A left parenthesis must follow the function name.`**

Explanation

A parenthesis must follow a function name in a specified expression.

060 **`A function name cannot be obtained: please supply it.`**

Explanation

While parsing a line number specification, a current function name could not be determined. An explicit function name specification is required.

061 **`The function-name / signal-name has been truncated to 8 characters.`**

Explanation

A **`hook`** command was issued, and the specified name has been truncated to 8 characters.

062 **`The config file can only be changed if the configuration is being saved.`**

Explanation

To change the configuration filename, the configuration save value must be 'Y'.

063 **`The [command] command cannot be specified in the profile.`**

Explanation

The command [command] is not supported in the profile. Commands that are supported include: **`abort`**, **`auto`**, **`break`**, **`config`**, **`define`**, **`disable`**, **`drop`**, **`enable`**, **`help`**, question mark (**`?`**) commands, **`ignore`**, **`install`**, **`log`**, **`on`**, percent (**`%`**) commands, **`query`**, **`set`**, **`system`**, **`trace`**, **`rsystem`**, **`undef`**, **`user`**, and **`window`**.

Action

Remove the command [command] from the profile.

064 **`Command's arguments, if any, are not processed.`**

Explanation

An entered command's arguments are ignored. This is usually because of the failure of the command to be processed.

066 **`End of profile/file: an incomplete ON command has been discarded.`**

Explanation

An incomplete **`on`** command was entered during debugger initialization.

067 `The ON command has been discarded.`

Explanation

An attempt to perform an **on** command failed.

068 `The [command] command can only be issued from a Rexx exec or Clist.`

Explanation

The **dbinput** and **dblog** commands can only be issued from an EXEC or CLIST.

069 `Prompt length may not exceed ''[prompt-length]'' characters.`

Explanation

A **dbinput** command was issued that specified a prompt, but that prompt string was longer than [prompt-length].

070 `The DBINPUT command gets input from the terminal when the debugger is prompting for terminal input.`

Explanation

A **dbinput** command has been encountered and the debugger is is now waiting for input.

Action

Enter data.

071 `Valid RESUME must be followed by no parameters, a function-name, a non-zero line-number, or both.`

Explanation

A **resume** command was issued with invalid parameters.

072 `Key definition text exceeds maximum allowed.`

Explanation

A **keys** command was entered but the definition text is greater than the limit.

074 `A TRANSFER command without arguments is ignored.`

Explanation

A **transfer** command was issued, but arguments are required.

076 `The ''signal''/''signals'' keyword can only be specified with IGNORE.`

Explanation

A **signal** keyword was specified on a **hook** or **query** command, but it is only supported on an **ignore** command.

077 `Invalid syntax specified. Valid syntax is ''IGNORE SIG.... signal''`

Explanation

An invalid **ignore** command was issued.

078 `Only an integer may follow a STEP or CONTINUE command.`

Explanation

A **step** or **continue** command was issued with a non-integer argument.

079 `Unable to open an input file to the terminal - DBINPUT fails.`

Explanation

A **dbinput** command was encountered, but the debugger was unable to open the terminal input file. **dbinput** ends without obtaining user input.

080 `The only valid prefix command is 'd' to drop a watch.`

Explanation

A prefix command other than **d** was entered on a line that had a watch. The only valid command is **d**[rop].

081 `A section-name cannot be greater than 7 characters.`

Explanation

A name was entered with a length greater than 7 characters.

082 `Out of memory - command(s) ignored.`

Explanation

During execution of a command, a storage allocation failed. The command does not execute.

083 `Out of memory - scrounging around, end ASAP.`

Explanation

An attempt to **malloc** storage failed. The debugger is now using a limited number of preallocated buffers.

Action

Do something that might free debugger storage or terminate the debugging session.

084 `Out of memory - debugger file cannot be read.`

Explanation

While reading a debugger file, an out of memory condition was experienced. The debugger file is unusable.

085 `An invalid symbol follows the EXIT command.`

Explanation

The **exit** command was entered with a keyword operand that was not **nodrop**. The **exit** command is ignored.

086 `An invalid keyword has been specified to the STORAGE command.`

Explanation

A **storage** command was entered with an invalid keyword.

087 `Unable to read from the terminal - DBINPUT fails.`

Explanation

A **dbinput** command was encountered, and the input terminal file was successfully opened, but the debugger was unable to read from the terminal. **dbinput** ends without obtaining user input.

088 `Missing/invalid LINESIZE value.`

Explanation

An **auto** command was issued with an invalid **linesize** value.

089 `Continue ON command list definition...`

Explanation

An **on** command has been entered, but is not complete. Enter the rest of the command.

090 **The expression specified is invalid.**

Explanation

While parsing an expression, invalid syntax was found.

091 **A ''='' must separate the source from the target in an ASSIGN command.**

Explanation

An **assign** command was issued with an inappropriate equals sign. The format is

```
ASSIGN var=value
```

092 **A '','' must separate the operands of the COPY command.**

Explanation

The operands to the **copy** command must be separated with commas.

093 **An invalid delimiter follows a valid command - the command is not executed.**

Explanation

A syntactically correct **assign**, **copy**, or **return** statement was entered with trailing characters.

094 **The syntax specified for the function call is not valid.**

Explanation

While parsing a builtin debugger function, invalid syntax was found.

095 **The string is too long or is not terminated.**

Explanation

While processing a token, either a string that was not terminated or a string that was too long was found.

096 **EXEC / CLIST variable name is longer than 256 characters.**

Explanation

While reading text from the input stream, an EXEC or CLIST name with a length greater than 256 characters was found.

097 **`The Termin window is not wide enough for the prompt - prompt truncated.`**

Explanation

A **`prompt`** command was issued, but the Termin window is not wide enough to display the prompt text. The prompt is truncated.

098 **`End of line number range cannot be less than the start.`**

Explanation

A line range specification was used where the end-of-range specification was less than the start-of-range specification.

099 **`Could not set desired scope.`**

Explanation

A **`scope`** command was entered, but the specified scope is not in the current call path.

100 **`# followed by a null / undefined macro name.`**

Explanation

While parsing an expression, a pound sign (#) was found, followed by an invalid macro name.

101 **`A format specification may not be specified with this TRANSFER keyword.`**

Explanation

A **`transfer`** command was issued with an invalid format. Acceptable formats include: *offset*, *address*, *value*, or *lineno.*

102 **`A LOG file has not been specified. Logging has not been turned on.`**

Explanation

A **`log start`** command was issued, but a log file had not been specified.

Action

Define a log file with the **`log file`** command.

103 **`Right bracket does not follow valid subscript.`**

Explanation

A right bracket was expected in an expression, but it was not found.

104 `Out of memory - overwriting debugger memory, end ASAP!!!`

Explanation

An attempt to **malloc** memory failed for a request that is larger than the preallocated debugger memory pool.

Action

Terminate use of the debugger now.

---

105 `Invalid text follows valid KEYS keyword.`

Explanation

A **keys** command was issued followed by invalid text.

---

106 `Invalid PF key number: valid numbers are 1-24.`

Explanation

A **keys** command was issued with an invalid PF key number. PF keys are limited to the range 1-24.

---

107 `Syntax is: KEYS DEFINE n ''text''.`

Explanation

A **keys** command was issued with an invalid syntax.

---

108 `Cannot drop a non-existent watch.`

Explanation

A **drop** command was issued for a line that does not have a watch.

---

110 `Invalid syntax for color/attribute specification.`

Explanation

A **window color** command was issued with an invalid color specification.

---

111 `The Command, Log, Source, and Status windows may not be closed.`

Explanation

A **window close** command was issued for a window that may not be closed.

---

112 `This command is valid only in full-screen mode.`

Explanation

A **browse** command was issued, while in line mode.

---

113 **`The WINDOW sub-command ``[subcmd]'' may not be specified in the profile.`**

Explanation

A **`window`** [subcmd] command was issued in the profile, but this subcommand is not supported from the profile.

114 **`Placeholder replacement not successful.`**

Explanation

This happens either during placeholder replacement while expanding a macro or from an **`expr`** command. Is is usually accompanied by **`LSCD124`**, **`LSCD125`**, or **`LSCD126`**.

115 **`Placeholder replacement not successful: valid expression not found.`**

Explanation

During placeholder replacement, an invalid expression was found.

116 **`Placeholder replacement not successful: input buffer overflow.`**

Explanation

During placeholder replacement, more space was needed than was available for expansion.

117 **`Commands to continue execution may not be issued when the [window] window is in this state.`**

Explanation

The command entered is ignored when the Termout or Termin window requires attention. Commands not supported include: **`continue`**, **`step`**, **`go`**, **`runto`**, **`resume`**, and **`goto`**.

Action

Resolve the pending terminal input or output event to be able to issue these commands.

118 **`This command may not be issued when the [window] window is in this state.`**

Explanation

The command entered is ignored when the Termout or Termin window requires attention. Commands not supported include: **`exec`**, and **`window off`**.

Action

Resolve the pending terminal input or output event to be able to issue these commands.

119 **`One or more missing right parentheses assumed to complete expression.`**

Explanation

While parsing an expression, it was found to be missing at least one right parenthesis.

120 **`This WINDOW sub-command may not be specified in the configuration file.`**

Explanation

A **`window`** command was found in a configuration file, but the specified subcommand can not be issued from this file.

121 **`Only KEYS/WINDOW commands may be specified in the configuration file.`**

Explanation

While reading the configuration file, a command other than the **`keys`** or **`window`** command was found.

122 **`This CONFIG sub-command may not be specified in the profile.`**

Explanation

The **`window config`** command was found while parsing the profile. Processing of the profile continues.

The **`config file`** command can be issued from a PROFILE; however, a **`window config`** command is not legal in a PROFILE.

124 **`The cursor is not in any window.`**

Explanation

A window command or text was entered, but the cursor was not in any window.

125 **`The cursor is not in valid window area.`**

Explanation

A window command or text was entered, but the cursor was not in a valid window area.

126 **`The extracted line is too long.`**

Explanation

The text entered was too long.

127 `A maximum of 40 watches is permitted.`

Explanation

A **watch** command was entered, but the maximum number of defined watches is 40.

128 `A format cannot be specified for ''dump'' style watches.`

Explanation

A **watch** command was issued that specified **dump**, but it also specified a format. This is not supported.

129 `This [sub-cmd] sub-command may only be specified in the profile.`

Explanation

A **window** [sub-cmd] command was issued, but the **intercepts**, **memory**, and **config** sub-commands may be specified only in the profile.

131 `The format specification is invalid or causes a buffer overflow.`

Explanation

While formatting a string, either an invalid format was found, or a buffer overflow occurred.

132 `The debugger is entered due to signal [signal], abend code [code].`

Explanation

An abend [code] in the program being debugged has occurred, causing signal [signal] to be received.

133 `RESUME may be issued only when the program is stopped at either a line hook or an epilog hook.`

Explanation

A **resume** command was issued, but this can be issued only on a function line or a function return, an exit, or at a line that a user-supplied break caused the program to stop.

134 `Cannot RESUME. [explanation]`

Explanation

A **resume** command was issued to resume in a function that is not in the current calling stack.

135 `Unable to locate line within specified function. Not RESUMEd.`

Explanation

A **`resume`** *function-name line-number* command was issued, but the specified line number does not appear in the specified function.

136 `Unable to locate line within current function. Not RESUMEd.`

Explanation

A **`resume`** *line-number* command was issued, but the specified line number does not appear in the current function.

137 `Writing to storage report output file [filename]. [errno]`

Explanation

An attempt to write a storage report record to file [filename] failed. [errno] will be non-blank if an error number was provided.

138 `Return code of SYSTEM [command]: [rc].`

Explanation

A **`system`** [command] command was issued, but the [command] returned a non-zero return code [rc].

139 `Debugger forcing program termination.`

Explanation

An error occurred during debugger initialization. The debugger terminates.

141 `There is insufficient memory for the debugger to start up.`

Explanation

During debugger initialization, it was determined that there is not enough storage to start the debugger. The debugger terminates.

142 `Debugger unrecoverable internal error.`

Explanation

The debugger encountered an internal error. The debugger terminates.

Action

Contact SAS/C Technical Support.

143 `Debugger recoverable internal error. Traceback follows:`

Explanation

The debugger encountered an internal error that it can recover from.

Action

Contact SAS/C Technical Support.

---

144 `Not enough functions in calling sequence. Enter WHERE to display traceback.`

Explanation

A transfer command or a scope command was issued but a function was specified that is not in the current calling segment. WHERE will show the current sequence.

---

145 `Save areas overlaid - traversal of calling sequence stopped.`

Explanation

An overlay was detected while traversing the call chain.

Action

Use the storage command to assist in locating the overlay.

---

146 `Cannot evaluate expression. [expression]`

Explanation

The specified expression [expression] is invalid, or multiple expressions were specified.

---

147 `WHATIS does not support this type.`

Explanation

A **`whatis`** command was entered for an unsupported type. **`whatis`** only supports: constants, arithmetics, classes, bit fields, enumerations, pointers, arrays, addresses, memory pointer, pointers to characters, references, and functions.

Action

---

148 `[message]`

Explanation

An error occurred during execution of a COPY, MONITOR, or WHATIS command. **`[message]`** describes the error.

---

149 **`Cannot create debugger temporary file.`**

Explanation

A file open failed for the debugger temporary file.

150 **`Debugger temporary file may have run out of space.`**

Explanation

While writing a debugger virtual page to the scratch file, an error occurred.

The debugger allocates a temporary file during debugger execution. When debugging large programs, it may be necessary to increase the size of this temporary file. On MVS, to increase the size of the file, before invoking the debugger, allocate a file with a ddname of SYSTMPDB. An example follows:

```
ALLOC F(SYSTMPDB) NEW SPACE(100) TRACKS UNIT(VIO)
```

On CMS, allocate a bigger read-write minidisk.

151 **`Cannot evaluate WHEN condition ``[condition]''. [message] Debugger will BREAK. ON commands, if any, are not executed.`**

Explanation

An invalid **`when`** condition has been encountered. Execution stops and any specified **`on`** commands are ignored.

152 **`Cannot find function ``[function]'' in file ``[filename]''.`**

Explanation

An attempt to find function [function] in file [filename] failed. This should not be possible with a complete debugger file.

Action

Contact SAS/C Technical Support.

153 **`Message buffer overflowed while processing expression.`**

Explanation

During the evaluation of an expression, the message buffer would not hold all message text.

154 **`No debugger macro variable defined for ``[macro-name]''.`**

Explanation

A **`define`** *macro-name* was issued, but there is not a macro named [macro-name].

155 `Macro redefinition. Old definition was: [old-macro-def].`

Explanation

A **`define`** *macro-name* "*new-macro-def*" was entered. *macro-name* already existed with a definition of [old-macro-def]. The previous macro definition is replaced by the specified definition.

156 `No debugger macro variable defined for ''[macro-name]''.`

Explanation

An **`undef`** *macro-name* was issued, but there is not a macro named [macro-name].

157 `TRANSFER not performed due to a memory access exception.`

Explanation

A **`transfer`** command failed due to a memory access exception.

158 `Cannot evaluate expression. [expression]`

Explanation

A **`transfer`** command was entered with an invalid expression specified.

159 `''[variable]'' is not a scalar: cannot TRANSFER.`

Explanation

A **`transfer`** command was issued, but the specified variable is not scalar.

160 `TRANSFER (DUMP / STR) expression should be of pointer type.`

Explanation

A **`transfer`**, **`variable`**, **`dump`**, or **`str`** command was issued, but expression specified was not a pointer.

161 `TRANSFER (DUMP / STR) string too long - insufficient memory.`

Explanation

A **`transfer`**, **`dump`**, or **`str`** command was issued, but the expression specified was too long.

162 `TRANSFER not performed: REXX/CLIST variable name ''[variable]'' or value ''[variable]'' is invalid.`

Explanation

A **`transfer`** command was issued, but either the specified REXX or CLIST variable name or the value is invalid.

163 `TRANSFER not performed: REXX/CLIST environment not active.`

Explanation

A **transfer** command was issued, but there was no REXX or CLIST exec running.

164 `TRANSFER not performed: environment does not have support for TSO CLIST variable interface.`

Explanation

A **transfer** command was issued, but the execution environment does not have support for the CLIST variable interface.

165 `TRANSFER not performed: (execshv) code [rc].`

Explanation

A **transfer** command was issued, but **execshv** returned a non-zero return code [rc].

166 `TRANSFER not performed. [explanation]`

Explanation

A **transfer** command was issued, but it was not performed, as indicated by [explanation]. Where [explanation] is one of the following strings:

- `''The variable name [var] was not found.''`
- `''' accompanied by LSCD0163`
- `''Not enough memory to complete operation.''`
- `''C library function execshv, internal error.''`
- `''''` (null string)

167 `Storage [storage_begin]:[storage_end] freed. Monitor(s) dropped:`

Explanation

The storage at location [storage_begin] through [storage_end] has been freed through the freeing of heap storage. All monitors that reference this storage are dropped.

168 `Load module [loadm] unloaded. Monitor(s) dropped:`

Explanation

The storage for a load module [loadm] has been freed as the result of the module being unloaded. All monitors that reference this storage are dropped.

169 `Cannot evaluate ''[expr]''. [desc]`

Explanation

A **monitor** command was issued, but the specified expression could not be evaluated because of [desc].

170 `This type of expression cannot be monitored.`

Explanation

A `monitor` command was issued specifying an expression that cannot be monitored. The expression must evaluate to an arithmetic, class variable, a bit-field, an enumeration, or pointer.

171 `The debugger profile ''[profname]'' execution failed, return code [profrc]. [errno]`

Explanation

The debugger profile [profname] returned a non-zero return code.

172 `System REXX interface failed. [errno-text]`

Explanation

A call to a REXX function failed because of an error in the REXX interaction.

173 `Cannot process EXEC command - not interactive.`

Explanation

The execution of an EXEC/CLIST was not attempted because it was not running in TSO.

174 `Cannot process EXEC or % command - not interactive.`

Explanation

The execution an EXEC/CLIST was not attempted because it was not running in TSO.

175 `EXEC command failed. Return code: [code]`

Explanation

On CMS, EXEC returned a non-zero return code [code].

176 `EXEC or % command failed. Return code: [code]`

Explanation

On MVS, an EXEC/CLIST returned a non-zero return code [code].

177 `Return code: [code]. Future invocation of CLISTs/EXECs may not work.`

Explanation

An error occurred during the execution of a CLIST/EXEC that may cause future invocations to fail.

178 `Line was too long and discarded: [text]...`

Explanation

During the fetch of a line to execute, it was discovered that the line was too long.

179 `Unable to open debugger input file.`

Explanation

An attempt to open the debugger input file (DBGIN) failed.

180 `Ddname ''[ddn]'' not allocated. Enter ''[ddn]'' data set names.`

Explanation

DDname [ddn] is not allocated, and you are prompted for the dataset name.

Action

Enter the corresponding data set name to be opened with this DDname.

181 `Ddname ''[ddn]'' not allocated.`

Explanation

DDname [ddn] was not allocated after the user prompt. Either an invalid data set name was specified or no input was specified.

182 `Unable to open debugger log file. See preceding runtime messages for details.`

Explanation

A **`log append`** command was entered, but for the log file failed to open.

183 `Cannot list source. [explanation]`

Explanation

A **`list`** command was issued, but the source could not be listed. The [explanation] should give a description of the problem encountered.

184 `Cannot scroll back [line-count] lines - scrolling to the top.`

Explanation

The debugger attempted to scroll back [line-count] lines, but that would scroll the source before the first line. The debugger stops scrolling at the top of the file.

185 `[line] line of range is set to 1.`

Explanation

A **`list`** command was issued with invalid start or end values. The described range value is set to 1. [line] is **`First`** or **`Last`**.

186 `No listing - last line of range ([last]) less than first line of range ([first]).`

Explanation

A **`list`** command specifying a range was issued but the last line was less than the first line. The last line of a range must be greater than the first line of a range.

187 `Source file ``[filename]'' is empty.`

Explanation

The source file [filename] did not contain any source.

188 `*** End of source ***`

Explanation

Indicates the normal end of the source code.

189 `End of source on line [linenum].`

Explanation

A **`list`** command was issued that specified a starting line greater than the last line of the file, [linenum].

190 `Ignoring member [member name] specified in DBGSRC definition.`

Explanation

The DDname DBGSRC is defined and a member name was specified in that definition. The member name will be ignored. Specification of a member name is not supported.

191 **`The debugger file for this compilation is not available - cannot list source.`**

Explanation

A **`list`** command was issued (perhaps implicitly on entry to a function), and the debugger file could not be found.

Action

If the debug source is accessible, use **`set search`** commands that locate the debug and source files. Then, issue the **`list`** command to indicate to the debugger that the debug file and the source file should be reloaded.

---

192 **`fopen unsuccessful. See preceding runtime messages for details.`**

Explanation

A file open was attempted, but failed. Other messages should indicate the cause of this problem. Preceding messages may not be present for HFS files.

---

193 **`Full-screen initialization failed: [expl].`**

Explanation

While processing a **`window on`** command, the initialization of the full-screen environment failed for reason [expl]. Line-mode will be used.

---

194 **`[hgt] window height is greater than the maximum height allowed for the window [window]. Set to [max].`**

Explanation

A **`config`** command was issued for a window [window], but a height [hgt] was specified that is greater than the maximum allowed for that window, and was reset to maximum [max].

---

195 **`[window] window height is less than minimum ([num]). Set to [num].`**

Explanation

When displaying a window an invalid height was entered. The minimum size is used.

---

196 **`[window] window row position is invalid for height specified. Set to [row].`**

Explanation

When displaying a window, the specified row would not allow all of the window to be displayed. Row [row] is used.

---

197 `[window] window width is greater than the maximum width allowed for the window ([wid]). Set to [max].`

Explanation

When displaying a window, a width was specified that was greater than allowed for this window. [max] is used as a width.

198 `[window] window width is less than minimum ([wid]). Set to [min].`

Explanation

When displaying a window, a width was specified that was less than allowed for this window. [min] is used as a width.

199 `[col] window column position is invalid for width specified. Set to [val].`

Explanation

A **`window config`** command was issued that specified a position that was invalid for the width of the window. An appropriate adjusted position [val] will be used.

200 `Zero / Multiple windows of this name are present - use <> to point.`

Explanation

A **`window`** command was issued that specified a window name, but there are either no windows with this name, or there are multiple windows with this name.

201 `Unable to reopen basic window. Continue debugging in line mode.`

Explanation

A **`window`** command command was issued, but the debugger was unable to open the window. Debugging will continue in line mode.

202 `This WINDOW cannot be OPENed/CLOSEd.`

Explanation

A **`window open`** or **`window close`** command was issued for a window that cannot be opened or closed. This includes the Command, Log, Source, and Status windows that may not be closed (or opened).

203 `Full-screen mode is on.`

Explanation

A **`window on`** command was issued, but this debugging session is already in full-screen mode.

204 **`Unable to determine initial working directory, ``.cdebugrc'' not processed.`**

Explanation

The directory for the "run command" file could not be found.

205 **`This WINDOW subcommand is not valid in this window.`**

Explanation

A **`window`** command was issued, but is not valid for the specified window.

206 **`Full-screen window open(s) failed - switching to line mode.`**

Explanation

During execution of a **`window open`** command, the opening of the window failed. Line-mode will be used.

207 **`Insufficient memory specified for [window] window - amount ignored.`**

Explanation

A **`window open`** command was issued for the [window] window, but the memory size for the window was not sufficient. [window] is Browse, Command, Log, or Source. The specified value is ignored.

Action

Specify a memory size of at least the height of the window * 255.

208 **`Window limit reached: cannot open another window.`**

Explanation

A **`window open`** command was issued but the maximum number of subwindows have already been opened.

Action

Close a currently open window.

209 **`[sev] detected by [mod]: [errcd].`**

Explanation

A [sev] (Error or Warning) was detected by module [mod].

210 **`Insufficient memory for the [window] window - unable to allocate [size] bytes.`**

Explanation

A **`window`** command for the [window] window was issued, but [size] bytes of storage could not be allocated.

211 **`Line too long.`**

Explanation

A request to write a line to the log was attempted, but the line is too long.

212 **`No continuation line.`**

Explanation

An attempt was made to write a line to the log that indicated that there was a continuation line, but no continuation line was found.

213 **`Line [num] is not in any function - the request is ignored.`**

Explanation

A command was issued that specified a non-existent line number [num].

214 **`Multiple ``r'' commands may not be specified.`**

Explanation

Multiple **`runto`** line commands were entered.

Action

Enter only one **`runto`** command at a time.

215 **`Both ``g'' and ``r'' prefix commands may not be simultaneously specified.`**

Explanation

The **`g`** (**`goto`**) and **`r`** (**`runto`**) prefix commands were specified simultaneously. This is not supported.

216 **`Unable to popup window to process error: ``[msg]''. Change is ignored.`**

Explanation

An attempt to display a popup window failed. The original message is [msg].

217 `Prefix-area commands may not be entered when no source is displayed.`

Explanation

A prefix-area command is entered when there is no source displayed.

218 `Length of input line is greater than 255 characters. It is truncated to 255 characters.`

Explanation

A line of input text is entered, but is longer than is supported.

219 `Input is not being prompted for at this time.`

Explanation

Text was entered in the Termin window, but no input was expected.

220 `Unable to open the termin window. Input intercept disabled.`

Explanation

An attempt was made to open the Termin window for input, but the Termin window could not be opened. Input will be taken from **`stdin`**.

221 `Unable to open the termout window. Output intercept disabled.`

Explanation

An attempt was made to open the Termout window for output, but the Termout window could not be opened. Output will be sent to **`stdout`**.

222 `Discarding input in [window] window.`

Explanation

Input text was entered in an output window. The input is discarded.

223 `Debugger file line-numbers do not correspond to source.`

Explanation

A command was issued, and in the evaluation of that command, it was determined that the debug and source files did not match. This many indicate that the source file has been modified since the last compilation.

224 `Section-name [section name] was not compiled DEBug (-d). There are no ''hooks'' in the code, and there is no debugger file available.`

Explanation

In an attempt to read the debugger file, it was determined that the code had not been compiled with the **`debug`** option. This file cannot be debugged with the debugger.

Action

Recompile the module, specifying the **`debug`** option.

---

225 `No debugger file - open failed, see runtime messages for details.`

Explanation

The debugger file was not found. Preceding messages may not be present when opening HFS files.

Action

If the modules were compiled using the **`debug`** option and the debug files are accessible, issue a **`set search`** command to identify the location of the files and the issue the **`list`** command.

---

226 `Debugger file ''[filename]'' is invalid and not used.`

Explanation

While reading a debugger file, it was determined that the file was invalid. The debugger file is not loaded.

Action

Recreate the debugger file.

---

227 `Debugger file ''[filename]'' is from Release 5.00 or before, which is not supported by this debugger.`

Explanation

While reading a debugger file, it was determined that the file was generated by a Release 5.00 or earlier compiler, which is not supported by this level of the debugger. The debugger file is not loaded.

Action

Recompile the source with a more current level of the the compiler.

---

228 **`Debugger file [filename] was not created in the same compilation as the object ([filename]).`**

Explanation

On CMS, while reading a cached debug file, it was determined that the debug file did not have the same timestamp as the object deck. No other debugger file exists. The file is not debuggable.

Action

Recreate or locate the current debugger file.

---

229 **`Debugger file for section name [section name] was not created in the same compilation as the object ([filename]).`**

Explanation

On MVS, while reading a cached debug file, it was determined that the debug file did not have the same timestamp as the object deck. No other debugger file exists. The file is not debuggable.

Action

Recreate or locate the current debugger file.

---

231 **`Dump terminated due to a memory access exception at ??.`**

Explanation

A **`dump`** command was issued, but a memory access exception occurred. Dumping stops.

---

232 **`Cannot evaluate ''[expression]''. [msg]`**

Explanation

Could not evaluate [expression]. The reason is stated in [msg].

---

233 **`Memory access exception at [address] while determining string length.`**

Explanation

A **`dump`** command was issued, and an access exception occurred when loading the string length.

---

234 **`''[operand]'' must be of address type.`**

Explanation

A **`dump`** command was issued, but the operand specified was was not an address type.

---

235 `A count may not be specified for an expression of this type.`

Explanation

A **print** command was issued with a count, but a count cannot be specified with this data type.

236 `Support for specifying a ''TAG'' type within parentheses after the expression to be printed has been dropped. Use casts to achieve a similar result.`

Explanation

In previous releases (prior to 5.50) it was possible to specify a **TAG** after an expression. Now, simply cast the value to the appropriate type.

237 `A ''COUNT'' may not be specified if the output is redirected to the Print window - the ''COUNT'' is ignored.`

Explanation

A **print** command was issued that specified a *count*. This is not supported when the output is being directed to the Print window. The *count* is ignored.

238 `Unable to access ''[var]'' due to a memory access exception.`

Explanation

During the access of the bit-field, pointer, function, arithmetic, array, or **enum** variable [var], an access exception occurred.

239 `A number must follow the '-' or '+'.`

Explanation

A **scope** command was issued with a **+** or **-** that was not followed by an integer.

240 `A maximum of 1000 C macro expansions per line are allowed.`

Explanation

More than 1,000 C macros were invoked in a single line.

241 `C macro expansion caused input buffer overflow.`

Explanation

Output from macro expansion caused an overwrite of the input buffer.

242 `Cannot load special character table - default values used.`

Explanation

An attempt to load the special character table (`l$dcst`) failed. A default of a one-to-one mapping is used.

243 `No debugger macro defined for [macro].`

Explanation

A debugger macro was invoked, but not found.

244 `A maximum of 1000 macro expansions per line are allowed.`

Explanation

More than 1,000 macros were invoked in a single line.

Action

245 `Debugger macro expansion caused input buffer overflow.`

Explanation

While expanding a macro, the resulting text caused an overflow of the input buffer.

246 `No Configuration file specified.`

Explanation

A **`config save`** command was issued, but no configuration file has been specified.

Action

Use the **`config file`** command to specify a configuration file.

247 `Open failed for [filename].`

Explanation

A **`storage`** or **`config save`** command was issued, but the output file [filename] could not be opened.

For **storage**, the file is designated as follows:

CMS: **cms:** *pgmname* **DBGSTG A**

OE: **ddn:DBGSTG**

MVS: if **ddn:DBGSTG** is defined, the **ddn** is used, otherwise, **dsn:userid.pgmname.DBGSTG** is used.

For **config save**, the configuration file is specified with the **config file** command.

On CMS, this filename is named DBCONFIG. On MVS, it is named either the specified filename or *first-level-qualifier*. CDEBUG.CONFIG(*name*).

248 **No requests in list.**

Explanation

A **drop all** or **drop last** command was issued, but there are no current requests to drop.

249 **No request applies specifically to this location.**

Explanation

An **enable** or **disable** command was issued, but there was nothing to modify.

250 **Multiple requests apply to this location. '[request]' not performed.**

Explanation

An **enable** (**e**) or **disable** (**d**) command was entered, but multiple requests apply to this location. The intended request must be specified.

251 **Request number [num] already disabled.**

Explanation

A **disable** command was entered, but the request number [num] had already been disabled.

252 **Request number [num] does not exist.**

Explanation

A **disable** or **drop** command was issued specifying a request number [num] that did not exist.

253 **`Last request already disabled.`**

Explanation

A **`disable last`** command was issued, but the last entry had already been disabled.

254 **`Request number [num] already enabled.`**

Explanation

An **`enable`** command was issued specifying a request number [num] that was already enabled.

255 **`Last request already enabled.`**

Explanation

An **`enable last`** command was issued, but the last request was already enabled.

256 **`[req] not installed. An identical request ([num]) already exists.`**

Explanation

A **`continue`** command was issued, but an identical request (request number [num]) already exists.

257 **`[req] not installed. A similar request ([num]) already exists. However, COUNT has been changed to [cnt].`**

Explanation

A **`continue`** command was issued, but request [num] already exists. The number of **`continue`** requests has been increased to [cnt].

258 **`Type of ``[variable]'' (target) is invalid for the [command] command.`**

Explanation

If [command] is **`assign`**, an attempt was made to give [variable] a value with the **`assign`** command; however, [variable] was not an appropriate type.

If [command] is **`copy`**, an attempt was made to copy to [variable] with the **`copy`** command. The [variable] must be an address.

259 **`A null list was specified: the [command] command will not change the target.`**

Explanation

If [command] is **`assign`**, an **`assign`** *aggregate-type-expression*=*value-list* command was issued, but *value-list* was empty. An empty *value-list* is not accepted.

If [command] is **`copy`**, a **`copy`** *destination*,{*list*} command was issued, but {*list*} was empty. An empty list is not accepted.

260 `Assignment of a list to a scalar is invalid.`

Explanation

An **assign** *aggregate-type-expression*=*value-list* command was issued, but *aggregate-type-expression* is a *scalar-type-expression*.

261 `Cannot ''[command]'' class/struct/union type to scalar type.`

Explanation

A [command] command was issued attempting to assign a **class**, **struct**, or **union** typed variable to a scalar variable.

262 `Can only assign a structure/union declared with the same tag in the same compilation.`

Explanation

An **assign** command was issued that specified structures or unions that were not declared with the same tag.

263 `Need a list or class/struct/union to initialize a class/struct/union.`

Explanation

An **assign** command was issued that specified either **class**, **struct**, or **union** but the assignment value was not a class, structure, or a union.

264 `Ignoring excess item(s) in list.`

Explanation

While processing in a list specified on an **assign** command, more items were found to be assigned than existed in the destination expression. The excess items are ignored, and command processing continues as as if the excess items were not present.

265 `Type of ''[expression]'' (source) is invalid for the [command] command.`

Explanation

A [command] was issued that specified an [expression] that is invalid for the command.

266 **`Need a list to initialize an array in a structure.`**

Explanation

An **`assign`** command attempted to assign a value to an array in a structure. A list must be specified.

267 **`Assigning a list to a scalar element of an array is not valid.`**

Explanation

An **`assign`** command attempted to assign a list to a scalar element. A scalar must be specified.

268 **`This version of the debugger does not support debugger files from Release [rel].`**

Explanation

A debugger file from Release [rel] was found, but is not supported.

Action

Use an appropriate level of the debugger, or recompile at the current level.

269 **`Memory access exception resulted when performing assignment / copy.`**

Explanation

The storage address specified on either the source or destination in an **`assign`** or **`copy`** command was invalid. It is possible that a storage overlay occurred for those sections of the **`assign`** or **`copy`** that did not cause the access exception.

270 **`Source and target sizes are unequal.`**

Explanation

A **`copy`** command was issued and the specified source and target are not the same size.

271 **`[target] type of aggregate is invalid - simple type or enum type should be specified.`**

Explanation

A **`copy`** command was issued with an invalid target, where [target] is **`Implied target`** or **`Target`**.

273 **`The RETURN command may not be issued at this kind of hook.`**

Explanation

A **`return`** command may not be issued at this line in the function.

274 **`A List may not be specified for a function returning a scalar.`**

Explanation

A **`return`** command was issued that specified a list, but this function returns a scalar.

275 **`The RETURN command may not be issued at a SIGNAL.`**

Explanation

A **`return`** command may not be issued at this time.

276 **`A Return value may not be specified for a function returning void.`**

Explanation

A **`return`** command was issued with a value specified. This is not allowed for a function that returns void.

277 **`No return value specified for function returning int. Return is performed.`**

Explanation

A RETURN command was issued that did not specify a return value, but this function should return an int. The return is executed using the value in register 15 as an undefined **`int`** return value.

278 **`A Return value must be specified.`**

Explanation

A **`return`** command was issued that did not specify a return value but a return value needs to be specified.

279 **`Cannot RETURN class/struct/union to a function returning a scalar.`**

Explanation

A **`return`** command was issued attempting to return a **`class`**, **`struct`**, or **`union`** when the function should be returning a scalar.

280 **`The return value is not used by the caller - the RETURN is not performed.`**

Explanation

Because the return value is not used, the **`return`** *value* is not performed.

281 **`No debugger file is available for function [function].`**

Explanation

There is no debugger file for this function.

Action

If the function is a user compilable function, recompile with debugging information enabled, and invoke the debugger again. If a debug file does exist, issue the **`set search debug=`** command followed by a **`list`** command to attempt to reload the debugger file.

---

282 **`fputs() error while writing to configuration file [filename].`**

Explanation

An error occurred while writing to the configuration file [filename].

---

283 **`A missing or invalid % format [format] is specified.`**

Explanation

A **`print`** command was issued that specified more than one format, but only one format is allowed.

---

284 **`Format specification may not be specified for structures, but is used for all fields in the structure.`**

Explanation

A **`print`** command was issued that specified a format specification. It is ignored.

---

285 **`The function has no parameters.`**

Explanation

A **`print`** command was issued that did not specify any operands at a function entry. The default **`print`** command at a function entry is to print the function parameter, but this function has no parameters.

---

286 **`Memory access exception resulted while accessing area to be monitored.`**

Explanation

During the evaluation of a **`monitor`** command, a storage access exception was recognized while referencing the area to be monitored. The monitor is not set.

---

287 `Long message has been truncated.`

Explanation

The debugger is attempting to print a message greater than its message buffer length. This is an internal error in the debugger. The message is truncated.

Action

Contact SAS/C Technical Support.

---

288 `Not running in interactive mode - help not available.`

Explanation

A **help** command was entered, but help is available only when running interactively.

---

289 `DBGHELP must be allocated for access to the help system. The debugger could not execute the L$DBHELP CLIST to automatically allocate DBGHELP.`

Explanation

A **help** command was entered, but the L$DBHELP CLIST did not complete successfully.

---

290 `The Help key (PF[num]) cannot be redefined using a KEYS DEFINE command; instead, use the KEYS HELP command.`

Explanation

The **keys define** command was issued attempting to set the Help key.

---

291 `[Command | Log] line has been truncated - lines in the [window] window cannot exceed 251 characters`

Explanation

A command was entered that exceeded 251 characters. Either the command line or a log line has been truncated.

---

293 `No entry for ''[text]'' in the help system.`

Explanation

A **help** command was issued, but [keyword] was not found.

---

294 `Terminal width ([term]) exceeds ''[max]'' columns. Problems may occur.`

Explanation

The width of the terminal exceeds [max]. Display results are not predictable.

---

295 `Unable to access the help system.`

Explanation

A **help** command was issued, but the help file could not be opened.

296 `The ISPF environment is not active.`

Explanation

A pfkey that is assigned to ISPF was pressed, but ISPF is not active.

297 `The Help key cannot be assigned to ISPF.`

Explanation

An attempt was made to assign the help key to ISPF. There must be a valid Help key.

Action

To assign the specified pfkey to ISPF, first reassign another, non-ISPF key to be the help key, then the specified key can be reassigned.

298 `A key assigned to ISPF cannot be made the Help key.`

Explanation

An attempt was made to assign the **help** function to a key already assigned to ISPF.

300 `Insufficient memory to hold macro name longer than [cnt] characters - the macro has been discarded.`

Explanation

While reading the preprocessor symbols, a macro name was encountered with a length greater than [cnt]. The macro cannot be used in the debugger session.

301 `The configuration has been written out to [filename].`

Explanation

A **config save** command was issued and the configuration was successfully written to file [filename].

302 `The debugger file [filename] is not complete.`

Explanation

While reading a debugger file, the end of file occurred before all sections of the debug file were found.

303 **`Source file has shrunk to [num] lines (from [num] lines) after the debugger first opened it.`**

Explanation

While trying to read a line from the source, an error occurred. This usually means that the source file has been replaced since the first time this source was loaded.

Action

Restart the debugger session. If this does not solve the problem, and the source was not replaced, then it is probably an internal debugger error.

---

304 **`More than ``[max]'' browse windows cannot be simultaneously open.`**

Explanation

Only [max] browse windows can be opened at the same time.

---

305 **`A debugger file is not available: cannot set breakpoints on lines in function [function].`**

Explanation

An attempt to open the debugger file was unsuccessful.

Action

Issue the appropriate **`set search debug`** command to locate the debugger files, and issue the **`list`** command to reattempt the loading of the debugger file.

---

306 **`Cannot set a breakpoint as function [function] is split across source files.`**

Explanation

The debugger does not support functions that start in one source file and end in a different source file.

---

307 **`Cannot set a breakpoint on line [num] as function [function] starts on line [start-num] and ends on line [end-num].`**

Explanation

An attempt was made to set a breakpoint at a line number outside the line-number range of the function.

---

308 **`Cannot set a breakpoint on line [num] of function ``[function]'' as there is no code generated for this line.`**

Explanation

An attempt was made to set a breakpoint at a line with no code.

---

309 **`Debugging a function in section-name [sname] ...`**

Explanation

Specifies the section name that was being debugged when the following errors were discovered. Other messages will be issued after LSCD309. Please see messages LSCD359 and LSCD363.

310 **`Forcing a break as the debugger could not install a line hook breakpoint. The request with the invalid line number should be dropped.`**

Explanation

A **`break`** *func nnn* command was entered, but the line did not exist. A dummy break was installed.

311 **`Could not open debugger startup file ``[filename]''. [errno_text].`**

Explanation

Could not open the "run command" file. Note that the file has different filenames according to operating system:

CMS: **`//cms: cdebug rc *`**

MVS: If DD DBGRC is allocated, then the DD is used; otherwise, **`dsn:`** *userid*`.cdebug.rc`

OE: **`//hfs:`***login-directory*`/.cdebugrc`

312 **`After a function name, only an integer and a ')' can follow a '('.`**

Explanation

The function name is incorrectly specified.

313 **`A function name specified within double qoutes is invalid.`**

Explanation

The function name is incorrectly specified. It is probably a C++ specification that is incorrect.

314 **`AUTO EXTNAME cannot be changed when the AUTO CXX is option is on.`**

Explanation

An **`auto extname`** command was issued, but the extended name support cannot be changed while the **`cxx`** option is on.

315 `Could not resolve function name ''[function]'' using merged debugger files.`

Explanation

The name that was specified was not found in any of the loaded routines.

316 `There is just one instance of function name [func] in the merged debugger files - the subscript is ignored.`

Explanation

A function name [func] was specified with a subscript, but there is only one function with that name. The subscript specification has no meaning.

317 `A function subscript of zero may not be specified in this command.`

Explanation

A function name was specified in a command, but a subscript of zero was also specified. Zero is not supported.

318 `Only C macros with no arguments are supported by the debugger.`

Explanation

Parms were specified with a C macro, but parms are not supported.

319 `Cannot access auto variables from a lexically enclosing function.`

Explanation

An attempt was made to reference an automatic variable from an invalid scope.

320 `Invalid type of expression for a reference type RETURN.`

Explanation

A **`return`** command was issued, but a reference type must be specified.

321 `Cannot access members from a lexically enclosing class.`

Explanation

An attempt was made to reference a number from an invalid scope.

322 `Using this particular instance of the class for monitor purposes.`

Explanation

While debugging a class member function, a **`monitor`** command specified a non-static member variable; for example, **`m`** of the same class as the "this" argument. The variable that is contained in the current "this" instance; for example, "this->m", is the one that is monitored.

323 **`The ASSIGN command may not be used with an object which has, or an object whose member has, a base class.`**

Explanation

An assignment can only be made to a base class.

324 **`Requests on all instances of an overloaded function may only be set on ''calls'', ''entry'', or ''return'' type hooks.`**

Explanation

An **`on`** command specified a hook of other than **`call`**, **`entry`**, or **`return`** (or **`*`**) for an overloaded function that is not supported.

325 **`The RETURN command may not be used to return an object which has, or an object whose member has, a base class.`**

Explanation

A **`return`** command was issued that specified an object, and either the object or a member of the object has a base class. This is not supported.

326 **`Memory access exception accessing virtual base class [class].`**

Explanation

An access exception occurred while processing a monitor event that attempted to access the virtual base class [class].

327 **`Out of memory - command output is not complete.`**

Explanation

An attempt to write a message failed, probably because of an out-of-storage condition. If this message was a result of a command, the command may or may not have executed successfully.

328 **`Classes are inherited: address/offset translation performed: [address] ──> [offset].`**

Explanation

A conversion between pointers or member pointers of different related classes was performed. This message documents the change in object addresses or in member pointer offsets.

329 **`The RETURN command may not be used to return an object which is returned using a copy constructor.`**

Explanation

A **`return`** command was issued that specified a copy constructor. This is not supported.

330 **`Invalid / inconsistent type of expression for a reference type RETURN.`**

Explanation

A **`return`** command was issued that specified a reference type that was inconsistent with the function definition.

Action

Look at the function definition.

331 **`Only a pointer to a member or zero can be assigned to a pointer to a member.`**

Explanation

An invalid source expression was supplied on an **`assign`** command.

332 **`Unable to perform the ASSIGN command. [reason]`**

Explanation

An **`assign`** command was entered, but could not be performed because of [reason].

333 **`The ''function_name:'' prefix is no longer supported. The SCOPE command may be used to provide similar, though enhanced, functionality.`**

Explanation

An old form of **`function_name:`** was used in an expression to provide for scoping, but this form is no longer supported.

334 **`Could not find function ''[function]'' in calling sequence. Enter WHERE to display traceback.`**

Explanation

A scope command was issued, but the specified functions were not in the calling sequence. The command **`where`** indicates the calling sequence.

335 `Could not perform the command as there is no log file defined.`

Explanation

A **`log capture`** or **`log file`** command was entered, but there is no log file.

336 `The current log file is [filename].`

Explanation

A **`log file`** command was entered. The current file is [filename].

337 `The log buffer is currently empty.`

Explanation

A **`log capture`** command was entered, but there are no lines to capture.

338 `Error writing log file. Output logging has been turned off.`

Explanation

An error occurred while writing to the log file. Writing to the log file is terminated.

339 `The WINDOW FIND command is not supported in the [window] window.`

Explanation

A **`window find`** command was issued for window [window], but this window does not support the **`find`** command.

340 `Cannot find search string: [str]`

Explanation

A **`window find`** command was issued, but the search string [str] was not found.

341 `Treating ``[name]'' as a C function name...`

Explanation

A hook was issued, using [name] as a C name.

342 ```[variable]'' is not in scope.`

Explanation

A **`browse`** command was issued, but the location specified was not addressable in the current scope.

343 **`A subscript after the function name is not valid in this context.`**

Explanation

A function name with a subscript was specified, but a subscript is not valid on this command.

---

344 **`The name of the file to be browsed has not been specified.`**

Explanation

A **`browse`** command was entered, but the no file has been specified.

---

345 **`A pointer to a member can only be assigned to a pointer to a member.`**

Explanation

An invalid target expression was supplied on an **`assign`** command.

---

346 **`The signal occurred at offset [address] which is greater than the size of the function ([address]).`**

Explanation

A signal occurred at an offset that is outside the function.

---

347 **`‘‘[expr]’’ is an expression of type void and cannot be PRINTed.`**

Explanation

A **`print`** command was issued that specified a value [expr] that evaluated to a type void. There is nothing to print.

---

348 **`Underflow in floating-point constant.`**

Explanation

During the conversion of a text value to a floating-point constant, an underflow occurred.

---

349 **`Overflow in floating-point constant.`**

Explanation

During the conversion of a text value to a floating-point constant, an overflow occurred.

---

350 **`Couldn’t find template ‘‘[template]’’ in the [name] list.`**

Explanation

A **`set search`** *file-tag*-**`"`***template*`"` command was issued, but the specified template, or templates, to remove were not found.

Action

Issue a **set search** *file-tag* **?** to see a list of the current templates for the named file, and check the specification of the template.

See *SAS/C Software: Changes and Enhancements, Release 6.50* for information about the **set search** command.

---

351 **An invalid conversion specifier has been specified in a template. Invalid text begins at ''[text]''.**

Explanation

In a **set search** command, a template was specified that contained the invalid conversion specifier [text].

Action

Correct the template specification. See *SAS/C Software: Changes and Enhancements, Release 6.50* for a description of search templates.

---

352 **A [specifier] conversion specifier must be followed by another conversion specifier.**

Explanation

In a **set search** command a **%l** or **%u** conversion specifier was used in a template, but it was not followed by another conversion specifier. The **%l** and **%u** conversion specifiers cause the replacement text case for the conversion specifier immediately following them to be converted to either uppercase or lowercase. As a result, they must always be followed by a second conversion specifier.

Action

Correct the template specification. See *SAS/C Software: Changes and Enhancements, Release 6.50* for a description of search templates.

---

353 **The ''[specifier]'' conversion specifier can only be used for a compilation on [system].**

Explanation

A conversion specifier of **%ddname** or **%membername** was specified in a template on a **set search** or **set cache** command, but the file was not compiled on the [system] that is being debugging. Where [system] is either MVS or CMS, the current template is skipped and processing continues with the next template.

Action

Use a conversion specifier appropriate for the compiling operating system.

---

354 **`The ''[specifier]'' conversion specifier cannot be used when there is no file name available.`**

Explanation

A defined conversion specifier was used in a **`set search`** or **`set cache`** command, but the original filename had not been specified. **`%sname`** is the only valid conversion specifier when the original filename is not specified. The current template is skipped and processing continues with the next template.

Action

Either define the original filename or use a section name.

---

355 **`Compilation with section name ''[section name]'' is pre-Release 5.00C and not supported by this debugger.`**

Explanation

The level of the compiler used with the executed code is not supported by this version of the debugger.

---

356 **`Source files compiled on [os1] cannot be debugged on [os2] without using SET SEARCH [type] commands to access files.`**

Explanation

When compiling on one operating system [os1] and debugging on another operating system [os2], because of file name convention differences, the **`set search`** [type] command should be used to indicate to the debugger how to correlate the names used in the compile to the names on the destination host.

Action

Issue the **`set search`** command to identify the files necessary for the debugger. See *SAS/C Software: Changes and Enhancements, Release 6.50* for information about the **`set search`** command.

---

357 **`This format of the debugger file name is not supported: [filename].`**

Explanation

Either the original filename [filename] could not be made into the CMS filename format, and a trace for the debug file had been enabled, or the [filename] has invalid opening and closing parentheses. The member name must be preceded by an open parenthesis, and followed by only one closed parenthesis. In either situation, the open file processing is terminated.

Action

None, unless this error is unexpected, then issue a **`set search debug ?`** command and examine the returned templates and original filename displayed in this message.

---

358 **`The template approach to search for files using the SET SEARCH and SET CACHE commands cannot be used with files compiled with this version of the compiler.`**

Explanation

The level of the compiler used to compile the source being debugged is incompatible with the use of templates in the definition of **`set search`** and **`set cache`** commands.

Action

If the debugger fails to find the files for debugging, then either recompile the modules using a more recent compiler, or make sure that files still reside in the same datasets as when they were compiled.

---

359 **`Tried to open [type] type file [where] ``[filename]''. [error-info].`**

Explanation

There are two possible explanations. A [type] filename, either the original filename or one that was generated from a template that was specified with the SET SEARCH [type] command or the SET CACHE (if [where] is "in cache") command, may not have been opened because of [error-info], and tracing for this file type had been enabled. Otherwise, an open of a [type] file was attempted and failed because of [error-info] and tracing for the file type had been enabled. In either case, processing continues with the next template.

Action

See Appendix 2, "ERRNO Values", for the various errno values that can be returned from file open if this error is not expected.

---

360 **`Debugger recoverable internal error. Inconsistent value: [value]. Traceback follows:`**

Explanation

An internal error has occurred in the debugger. The value [value] is inconsistent with earlier debugger logic. The requested function is not executed. An execution traceback follows.

Action

Contact SAS/C Technical Support.

---

361 **`Debugger recoverable internal error. Inconsistent values: 1: [num], 2: [num]. Traceback follows:`**

Explanation

An internal error has occurred in the debugger. The value [values] are inconsistent with earlier debugger logic. The requested function is not executed. An execution traceback follows.

Action

Contact SAS/C Technical Support.

---

362 **`The ''[specifier]'' conversion specifier can only be used in templates for [type] type files.`**

Explanation

When [type] is not **`userinclude`**, the conversion specifier of **`%ddname`** can only be specified for user include templates in the **`set search userinclude`** statement. Command processing ends.

Action

Specify a valid template. See *SAS/C Software: Changes and Enhancements, Release 6.50* for information about the **`set search`** command.

---

363 **`[type] type file ''[filename]'' found.`**

Explanation

A filename generated in a template specified for the file type [type] in a **`set search`** command was successfully opened and tracing was enabled for this type. File processing continues.

Action

If a successful open was unexpected for this template, use **`set search`** *file-tag* **`?`** determine the current template, and either change the template or rename the associated files to correspond to the currently defined template.

See *SAS/C Software: Changes and Enhancements, Release 6.50* for information about the **`set search`** command.

---

364 **`Tried to open DEBUG type CACHE file ''[filename]''. [errno_text]. The debugger will not use the cache mechanism for this compilation.`**

Explanation

A cache filename had been specified, but could not not be created.

Action

Specify a valid cache name. See *SAS/C Software: Changes and Enhancements, Release 6.50* for information about the **`set cache`** command.

---

365 **`Tried to write to DEBUG type CACHE file ''[filename]''. [expl]. The debugger will not use the cache mechanism for this compilation.`**

Explanation

While attempting to copy the debugger file to cache, an error occurred due to [expl]. The cache file will not be used for this debugger file.

---

366 `Tried to remove ``[parts]''. [error-info].`

Explanation

While reading in a debugger file, an error occurred that caused the debugging information to be considered invalid, but the attempt to free the parts of the file that had been read failed.

367 `The time-stamp of the debugger file ([filename]) in CACHE is different from the time-stamp of the object code ([filename]). The cache file will not be used.`

Explanation

While reading in a debug file from cache, it was determined that the date of the object code was different from the cache and a non-cached debugger file exists, so it will be used instead of the cached version. The cached debug file is ignored.

368 `The debugger file in ``[filename]'' will be copied to the CACHE location ``[location]''.`

Explanation

A debugger file was opened, and a cache file is defined, so the debugger file will be copied to the cache.

369 `Debugger unrecoverable internal error. Inconsistent value: [value].`

Explanation

An internal error has occurred in the debugger. The value [value] is inconsistent with earlier debugger logic. The requested function is not executed. An execution traceback follows.

Action

Contact SAS/C Technical Support.

370 `Debugger unrecoverable internal error. Inconsistent values: 1: [num], 2: [num].`

Explanation

An internal error has occurred in the debugger. The value [values] are inconsistent with earlier debugger logic. The requested function is not executed. An execution traceback follows.

Action

Contact SAS/C Technical Support.

371 `Debugger unrecoverable internal error. Inconsistent values: 1: [num], 2: [num], 3: [num].`

Explanation

An internal error has occurred in the debugger. The values are inconsistent with earlier debugger logic. The requested function is not executed. An execution traceback follows.

Action

Contact SAS/C Technical Support.

---

372 `The target of the [command] command, ''[target]'', is not in the address space of the program.`

Explanation

The specified variable [target] is not addressable in this address space. The [command], **`assign`** or **`copy`**, can only be used to modify variables in the current address space.

---

373 `The debugger is terminating due to lost communication connection with remote subject program.`

Explanation

The debugger has lost communications with the program being debugged. The debugger exits, and the state of the program being debugged is unknown.

---

374 `PRINT not performed due to a memory access exception.`

Explanation

While formatting a string, a memory access exception was encountered.

---

375 `The RSYSTEM command is not supported by the environment of the program being debugged.`

Explanation

This environment does not support the remote debugging system.

Action

See *SAS/C Software: Changes and Enhancements, Release 6.50* for more information on the use of remote debugging.

---

376 `The Debugger command ''[command]'' failed with return code [rc].`

Explanation

A debugger command was issued, but failed with return code [rc].

---

377 `RSYSTEM command ''[command]'' return code is [rc].`

Explanation

A remote debugger command [rcmd] was issued, but returned a non-zero return code [rc].

378 `Unable to set default search templates.`

Explanation

During debugger initialization, a failure occurred while setting the default **`set search`** templates.

379 `No executable profile ''[filename]'' found.`

Explanation

For OpenEdition, no debugger profile was found. The filename for OpenEdition is **`.cdebug`**.

380 `No transaction name specified.`

Explanation

No transaction name was specified in the command line. The transaction name for the program to be debugged must be specified after 'DBUG' in the command line.

Action

Run the debugger again, specifying the transaction name that is associated with the program to be debugged.

381 `The transaction name [transaction] is invalid or has not been defined to CICS.`

Explanation

The transaction name given in the command line could not be found.

Action

Make sure that the transaction name is defined to CICS and is spelled correctly. You may be able to issue the following CICS command to verify that the transaction has been defined:

CEMT INQUIRE TRANSACTION([transaction])

382 **`An EXEC CICS INQUIRE TRANSACTION [transaction] command failed with response code [code].`**

Explanation

CICS has returned an unexpected response code to an EXEC CICS INQUIRE TRANSACTION command. Some of the possible response codes include INVREQ and NOTAUTH.

Action

For the response code diagnostics and further explanation, see IBM publication *CICS System Programming Reference*. If the response code is NOTAUTH, then you may have to contact your site's CICS system administrator to obtain authority to issue this command.

---

383 **`The program [program] has not been defined to CICS.`**

Explanation

The program that is associated with the transaction that you are trying to debug has not been defined to CICS.

Action

Make sure that the program name is defined to CICS and is spelled correctly, and that the transaction definition points to the correct program name. You may be able to issue this CICS command to verify that the program has been defined:

CEMT INQUIRE PROGRAM[program]

Otherwise, contact your CICS system administrator for further assistance.

---

384 **`The program [program] is disabled.`**

Explanation

The program that is associated with the transaction that you are trying to debug has been marked as 'disabled' by CICS.

Action

To make the program available for use, you may be able to issue the following CICS command:

`CEMT SET PROGRAM [program] ENABLE`

Otherwise, contact your CICS system administrator for further assistance.

---

385 **`The program [program] can not be loaded by CICS.`**

Explanation

The program that is associated with the transaction that you are trying to debug has been correctly defined, but CICS is unable to load the program.

Action

Make sure that the program exists in a dataset that appears in the CICS DFHRPL library concatenation. Contact your CICS systems programmer for further assistance.

386 **`An EXEC CICS LOAD command for program [program] failed with response code [code].`**

Explanation

The program that is associated with the transaction that you are trying to debug failed to load.

Action

See the IBM CICS documentation for the response code diagnostics. If the response code is NOTAUTH, then a resource security check has failed. Contact your CICS system administrator for further assistance.

387 **`Application has not been linked with a SAS/C Release 6.00 or higher resident library.`**

Explanation

The program cannot be debugged because it was link-edited with an older version of the SAS/C resident library that does not support debugging under CICS.

Action

Relink the program to be debugged with a 6.00 or higher release of the SAS/C resident library, and then rerun the debugger.

388 **`EXEC CICS SEND MAP [map] failed with response code [code].`**

Explanation

The SAS/C debugger front-end transaction attempted to issue an EXEC CICS SEND MAP command that failed with the indicated response code.

Action

See IBM publication *CICS Application Programming Reference* for the response code diagnostics. Contact your CICS system administrator for further assistance.

389 **`An 'EXEC CICS SEND TEXT FROM [message]' command failed with response code [code].`**

Explanation

The SAS/C debugger front-end transaction attempted to issue an EXEC CICS SEND TEXT command to display an error message and the command failed with the specified CICS response code.

Action

See IBM publication *CICS Application Programming Reference* for the response code diagnostics. Contact your CICS system administrator for further assistance.

---

390 **`[file-tag] type files: TRACE is [status] and the search list is empty.`**

Explanation

In response to a **`set search`** *file-tag* **`?`** command, if the list of file templates is empty, then this message will be displayed. Otherwise, see LSCD391, LSCD392, and LSCD393. The TRACE is either **`on`** or **`off`**.

---

391 **`[file-tag] type files: TRACE is [status] and the search list contains:`**

Explanation

In response to a **`set search`** *file-tag* **`?`** command, if the list of file templates is not empty, then this message will be displayed followed by a LSCD392 or LSCD393 message (if [file-tag] is debug), then a series of LSCD392 messages. The TRACE is either **`on`** or **`off`**.

---

392 **`(There is no CACHE specified.)`**

Explanation

In response to a **`set search debug ?`** command, if a cache file has not been specified, this message will be displayed. See LSCD391, LSCD393, and LSCD394 for more information.

Action

If desired, a cache file can be specified with cross-development debugging using the **`set cache`** command. See *SAS/C Software: Changes and Enhancements, Release 6.50* for information on specifying a cache.

---

393 `(CACHE is ''[filename]''.)`

Explanation

In response to a **`set search debug ?`** command, a cache file has been specified as [filename]. See LSCD391, LSCD392, and LSCD394 for more information.

Action

If desired, the cache file can be eliminated with the **`set cache=`** command. See *SAS/C Software: Changes and Enhancements, Release 6.50* for information on specifying a cache.

---

394 `''[template]''`

Explanation

In response to a **`set search`** *file-tag* **`?`** command, if the list of file templates is not empty, then a series of these messages will be displayed preceded by message LSCD392.

Action

Templates can be modified, added, or removed with the **`set search`** *file-tag* **`=`** | **`+`** | **`-`** **`"`***`template`***`"`** command.

---

395 `[string]`

Explanation

A **`dblog`** command was encountered specifying the displayed [string].

---

396 `[macroname] [macrotext]`

Explanation

A **`define *`** or **`define`** [macroname] command was entered. [macroname] represents the defined macro name, and [macrotext] represents the replacement text for that macro.

---

397 `No debugger macro variables defined.`

Explanation

A **`define *`** command was entered to display all defined macros, but there are no macros currently defined.

---

398 `break on [name1] - context of [name2].`

Explanation

A hook was encountered but in another context.

---

399 `Dimension of array ``[array-name]'' is not known.`

Explanation

`sizeof`([array-name]) was specified in an expression, but the dimension of that array is not known.

---

400 `Cannot apply sizeof to [name] type ``[type]''.`

Explanation

`sizeof`([name]) was specified in an expression, but the [name] specified was of [type] **`function`** or **`bitfield`** which is not supported with **`sizeof`**.

---

401 `Type of ``[type]'' is invalid for the ``[oper]'' operation.`

Explanation

An expression was specified using an operand of type [type], but that type cannot be used with the [oper] operation.

---

402 ```[name]'' is not an lvalue.`

Explanation

[name] was specified in an expression, but it cannot be used as an lvalue.

---

403 `Cannot take the address of ``[name]''.`

Explanation

[name] was specified in an address expression, but it is a constant or the size of an object, and an address cannot be generated.

---

404 `Cannot apply & to the bit-field / enumeration constant ``[name]''.`

Explanation

[name] was specified in an address expression, but it is a bit-field or an enumeration, and an address cannot be generated.

---

405 `Computational error involving the ``[oper]'' operations with operands ``[operand1]'' and ``[operand2]''`

Explanation

During the evaluation of the [oper] operation, an exception occurred involving operands [operand1] and [operand2].

---

406 `Subtraction cannot be performed: pointers ``[ptr1]'' and ``[prt2]'' point to objects of different sizes ([size1] and [size2]).`

Explanation

A subtraction operation was specified using two pointers, [ptr1] and [ptr2], but they address objects of differing sizes [size1] and [size2]. This is not supported.

407 `Types of ``[var1]'' and ``[var2]'' are invalid for subtraction.`

Explanation

A subtraction operation was specified using two values, [var1] and [var2], but they are of types that do not support subtraction.

408 `The only constant that can be compared with a member pointer is 0.`

Explanation

A comparison operation was specified using a non-zero constant, but a pointer can only be compared to 0.

409 ```[name]'' is not a scalar.`

Explanation

A logical AND or OR expression was specified, but one of the specified operands was not a scalar.

410 `Type of ``[oper1]'' and/or ``[oper2]'' is invalid for the ``[operation]'' operation.`

Explanation

A binary expression was specified using operands [oper1] and [oper2], but the type of one of the operands is not vaid for the [operation] operation.

411 `One of ``[oper1]'' and ``[oper2]'' should be of pointer type and the other of integer type.`

Explanation

A pointer addition expression was specified, but one of the specified operands should be a pointer and one should be an integer.

412 `Only an identifier can precede the ``::'' operator.`

Explanation

Something other than a name precedes the ::.

413 `Cannot cast non-scalar type ``[name]''.`

Explanation

The debugger does not support casting of non-scalars.

414 `Casting to [type] is not allowed.`

Explanation

The debugger does not support casting to [type].

415 `"[var] is not a pointer to a member.`

Explanation

A `.*` or `->*` was specified in an expression, but the pointer specified was not a pointer to a class member.

416 `The operator ``.*'' or ``->*'' is invalid for ``[var]''.`

Explanation

A `.*` or `->*` was specified in an expression, but the object addressed is not a class member.

417 `Memory access exception accessing virtual function table pointed to by ``[var]''.`

Explanation

A `.*` or `->*` was specified in an expression, but the object address [var] is not valid.

418 `Pointer to member, ``[var]'', is NULL.`

Explanation

A `.*` or `->*` was specified in an expression, but the object address [var] is NULL.

419 `The [operator] operator cannot be applied to ``[var]''.`

Explanation

A `.` or `->` was specified in an expression, but the operation is not valid for the specified variable.

420 `Only one level of ``::'' following a ``.'' or ``->'' is supported.`

Explanation

A `.` or `->` was specified in an expression, with multiple `::` operators, but only one is supported.

421 `Only an identifier may follow a ``.'', ``->'', or ``::''.`

Explanation

A `.`, `->`, or `::` was specified in an expression, but the operand following the given operation was not an identifier.

422 `Memory access exception dereferencing pointer to member ``[member]''.`

Explanation

During an attempt to convert a pointer to a member, an access exception occurred that indicates an invalid pointer.

423 `Memory access exception accessing reference ``[var]''.`

Explanation

During an attempt to convert an object to the referenced type, an access exception occurred indicating an invalid pointer.

424 `Cannot obtain the ``value'' of ``[var]'' as it is not an lvalue.`

Explanation

The specified variable [var] does not have a value appropriate for fetching. It must be lvalue capable.

425 `Memory access exception accessing ``[var]''.`

Explanation

During an attempt to obtain the actual value of [var], an access exception occurred indicating an invalid pointer.

426 `No user commands have been installed.`

Explanation

An `install *` command was issued to list all user commands, but none were found.

427 `Listing source from [file]`

Explanation

Line mode only. Indicates the file where the source is coming from, but none were found.

428 `Resuming execution in function [func].`

Explanation

A **`resume`** function command was issued and execution is resumed in function [func].

429 `Resuming execution in function [func].`

Explanation

A **`resume`** *function-name lineno* command was issued and execution is resumed in function [func].

430 `MONITOR not installed, MONITOR [mon] already exists.`

Explanation

A **`monitor`** command was issued when monitor [mon] already exists for the specified variable.

431 `''[mon]''`

Explanation

A monitored variable was modified and this message is printed as a result of recognizing that monitor event. This message is sometimes being present:

- `Monitor hit: [Monitor Number] *`
- `. MONITOR [varname] (inactive)`
- `'' '' (recursive)`
- `'' '' LIBRARY IN ([name]) PRINT WHERE`

432 `[bitfield_n] was [bitfield_o]`

Explanation

A monitored bitfield was modified with a new value of [bitfield_n] and an old value of [bitfield_o]. If the bitfield was previously initialized, a tag of **`: uninitialized`** will be appended to [bitfield_o]. Note also that if an access exception occurs while examining the bitfield, the word IMPOSSIBLE will appear.

433 `[dumped_n] was [dumped_o]`

Explanation

A monitored field was modified with a new value of [dumped_n] and an old value of [dumped_o].

434 `[value_n] was [value_o]`

Explanation

A monitored field was modified with a new value of [value_n] and an old value of [value_o].

435 `Pointer in class [class] to virtual base class [baseclass] has been corrupted.`

Explanation

The internal pointer that is used to implement access to a virtual base has changed. This pointer never changes in the normal life of an object.

436 `base | virtual base ...printed above....`

Explanation

These are used when printing out a base class as part of a containing class.
When printing a normal base, you get:
`base <tagname> {<base contents>}`
When printing a virtual base:
`virtual base <tagname> {<base-contents>}`
If the virtual base was already printed, you get:
`virtual base <tagname> ...printed above...`

437 `base | virtual base ...printed above....`

Explanation

These are used when printing out a base class as part of a containing class.
When printing a normal base, you get:
`base <tagname> {<base contents>}`
When printing a virtual base:
`virtual base <tagname> {<base-contents>}`
If the virtual base was already printed, you get:
`virtual base <tagname> ...printed above...`

438 `Command not processed as there are [cnt] functions named [func]:`

Explanation

A function name was specified without a subscript, but there are [cnt] functions with name [func], a subscript must be specified.

Action

Reissue the command selecting the correct instance of the function.

439 `[fi] [func]`

Explanation

Displays the various instances of a particular function [func] along with the indices ([fi])to use when referencing a function instance with the debugger.

440 `PF key definitions:`

Explanation

A **`key list`** command was issued. This is the heading for the list of pfkey definitions.

441 `[pfnum] [action]`

Explanation

A **`key list`** command was issued, and [action] is the action when pf key[pfnum] is pressed.

442 `The current configuration file is [filename].`

Explanation

A **`config file`** command was issued, and the new configuration file is [filename]

443 `line [num]: no suffix allowed for 'g'`

Explanation

A prefix command **`g`** was entered for line [num] with a suffix, but suffixes are not allowed on the **`g`** prefix command.

444 `line [num]: invalid prefix command`

Explanation

A prefix command was entered for line [num], but it was not valid.

Action

Valid prefix commands are: **`b`** (**`break`**), **`d`** (**`disable`**), **`e`** (**`enable`**), **`g`** (**`goto`** and **`resume`**), **`q`** (**`query`**), **`r`** (**`runto`**), and **`t`** (**`trace`**).

445 `module not in calling sequence MODULE: [mod] [num]`

Explanation

A prefix command was entered with a module name, but the module [mod] is not in the current calling sequence.

446 `invalid line number LINE: [num]`

Explanation

A prefix command was entered with a line number [num] that is out of the range for this module.

---

447 `Invalid [val] value. Enter value:`

Explanation

An invalid character value [val] was entered.

Action

Enter a valid value.

---

448 `Invalid [val] value. Enter value:`

Explanation

An invalid decimal value [val] was entered.

Action

Enter a valid value.

---

449 `Formats begin with a '%' FORMAT [format]:`

Explanation

An invalid format was entered.

---

450 `Invalid hex value. value [val]:`

Explanation

An invalid hexadecimal value [val] was entered.

---

451 `invalid expression specified EXPR [expr]:`

Explanation

A **when** expression was entered that was invalid.

---

452 `Set system breakpoint at [location] to activate the ESCAPE command.`

Explanation

This message is issued at debugger initialization and indicates that a system breakpoint can be set at the specified location to enable the **escape** command.

---

453 `C-macro ``[orig]'' replaced by ``[repl]''.`

Explanation

The C-macro [orig] has been replaced with [repl].

454 `An EXEC CICS RECEIVE MAP [map] command failed with response code [code].`

Explanation

The SAS/C debugger front-end transaction attempted to issue an EXEC CICS RECEIVE MAP command and the command failed with the specified CICS response code.

Action

See the IBM CICS documentation for the response code diagnostics. Contact your CICS system administrator for further assistance.

455 `A putenv() for some debugger environment variable failed.`

Explanation

The SAS/C debugger front-end transaction calls putenv() to save the environment variables that are used by the remote debugger. One of these calls failed.

Action

The run time library should have produced a diagnostic message on the stderr transient data queue that describes the error condition that caused the putenv() to fail. Locate that message and proceed accordingly.

456 `[command] ([length]) as [exec]`

Explanation

Command [command] has been replaced with [exec].

Appendix 1
Library ABEND Codes

This appendix lists all ABENDs issued by the C run-time routines. Most of the ABENDs described here result from user errors. Storage ABENDs (messages 1200–1203 and 1205–1209) are generally caused by buffer overflows or overlays. The application can be tested with the **=storage** run-time option to get a storage corruption report. Refer to the *SAS/C Debugger User's Guide and Reference, Third Edition*, for a detailed explanation of the storage report and guidance on using the debugger to isolate storage problems.

If you suspect a problem with the product, contact the SAS Software Representative at your site.

The ABEND codes currently defined are:

1200 issued if corruption of **auto** storage control blocks is detected while a DSA is being freed.

1201 issued if corruption of **auto** storage control blocks is detected during allocation of a new DSA.

1202 issued if, during termination of a program run with the **=minimal** run-time option, it is discovered that the program has stored past the end of the automatic storage stack.

1203 issued if corruption of **auto** storage control blocks is detected while freeing all automatic storage at the end of program execution.

1204 issued if the target of a **longjmp** cannot be found. This may be caused by corruption of the target **jmp_buf** or by attempting to jump to a routine that is no longer active.

1205 issued if corruption of heap storage control blocks is detected during execution of **calloc/malloc**.

1206 issued if corruption of heap storage control blocks is detected when **calloc/malloc** attempts to reuse a free portion of the heap.

1207 issued if corruption of heap storage control blocks is detected by **free** during the release of a tract of heap storage.

1208 issued if the pointer to a block to be freed is found to be invalid by **free**. This may be caused by an overlay of storage near the block or by passing a pointer to an area not directly allocated by **calloc/malloc**.

1209 issued if corruption of heap storage control blocks is detected during program termination.

1210 issued when the **SIGABRT** signal is raised and default handling is in effect. **SIGABRT** can be raised by a call to **abort** or by using **raise** or **siggen**.

1211 issued if the argument to **unloadm** does not address a function in a module previously loaded by **loadm**.

1212 issued if required transient run-time functions cannot be loaded. Under MVS, the transient library must have been installed into the system link list or be defined as part of STEPLIB or CTRANS. Under CMS, the transient library must be present on an ACCESSed disk, in nucleus extensions, or in a segment. Under the OpenEdition shell, the transient library must be installed into the system linklist or be defined to the

library using the **STEPLIB** or the **ddn_CTRANS** environment variables.

For CICS applications, verify that CICS library support customization has been accomplished. Refer to the *SAS/C Installation Guide* for details on installing SAS/C CICS functions.

1213 issued if a subordinate C load module is abused by calling it directly (for example, via EXEC PGM=) as if it were a main program.

1214 issued if more than one recursive error occurs during diagnostic message processing. This is probably a run-time library error; the most likely cause is a malformed diagnostic message text.

1215 issued under CMS if an error occurs while trying to delete the storage occupied by a dynamically loaded module during a call to **unloadm**. The most likely cause is a storage overlay.

1216 issued under CMS if an invalid library control block is encountered during a call to **unloadm**. This is probably caused by a storage overlay.

1217 issued under CMS if an invalid library control block is encountered during a call to **loadm**. This is probably caused by a storage overlay.

1218 issued under CMS if an error occurs while trying to delete a buffer during dynamic loading. The most likely cause is a storage overlay.

1219 issued if the interface between the debugger and the library detects an error in its own processing or in the debugger. This can occur if the program overlays debugger data areas.

1220 issued if the **abort** debugger command is executed. It also can happen if multiple attention interrupts occur and the debugger is unable to respond normally due to a program loop or system slowdown.

1221 issued under CMS if an error is detected by CP while loading a library segment. It is probably an indication of a CP system error.

1222 issued if the library framework manager detects an error in its use. (The framework manager is a library component used as a subroutine by the debugger, interlanguage communication applications, and the CMS REXX function package and SUBCOM support.) This is most likely a library error, but it can be caused by overlay of library control blocks.

1223 used under TSO to indicate a system problem in the TSO SUBCOM interface. This is most likely a library or TSO error. Under CMS, this code is used to indicate that the nucleus extension used to manage sharing of the C run-time library has been dropped by the user, which is not permitted.

1224 issued when **longjmp** or **exit** attempts to terminate a non-C routine. This can occur when the **indep** option is used to mix the C language with another language or when the CMS REXX function package support is used.

1225 issued when there is no handler for an asynchronous signal, such as **SIGALRM** , for which the default action is termination. The default handling is to terminate the program.

1226 issued if the Full-Screen Support Library detects an internal error. This may be caused by an overlay of FSSL control blocks.

1227 issued when a library error occurs terminating a coprocess.

1228 issued if there is not enough storage available to extend the internal queue of IUCV or socket communication messages. The probable cause is that the number of incoming interrupts is greater than the program can handle.

1229 This is a CICS-only ABEND code that is issued when there is not enough storage available for an unconditional GETMAIN.

1230 issued when a POSIX function cannot be executed because OpenEdition is not installed or active, but the function does not have any defined error conditions.

1231 No TWA defined for a CICS Indep Application.

1233 issued if an internal error ocurred in the interlanguage communication support routines. This is most likely a library error, but could be caused by an overlay of library control blocks. *Please record any preceding messages and the value in register 15 at the time of ABEND, if possible, before calling SAS Institute's Technical Support Department.*

1234 issued if a call is made from a C program to a routine written in another high-level language or from a program in another high-level language to a C function, but the framework for the called language has not been created.

1235 issued for an interlanguage communication feature usage error. The error is explained by a preceding message.

1239 This is a CICS-only ABEND code that is issued when an unexpected internal error is encountered during abnormal termination processing.

1240 issued if an unexpected program check occurs during library ABEND analysis. The analysis routine is careful not to depend on the contents of run-time control blocks, so an occurrence of this ABEND probably indicates an oversight in the coding of the analysis routine as well as a storage overlay.

Appendix 2
ERRNO Values

This appendix lists all ERRNO values issued by the SAS/C run-time routines. The external **int** variable **errno** contains the number of the most recent error or warning condition detected by the run-time library. To use this value, include the header file **<errno.h>**.

If no error or warning condition is detected, the value of **errno** is 0. After program execution starts, **errno** is never reset to 0 by the library. Programs that use **errno** for information about unusual conditions must set it to 0 before calling a library routine that may detect such a condition.

The **<errno.h>** file contains declarations of the **errno** variable and definitions of symbolic names for the values that can be assigned. These names rather than numeric values should be used for **errno**.

The ERRNO values currently defined are:

E2BIG	argument list for **exec** function too large
EACCES	inaccessible socket or permission denied
EADDRINUSE	socket address already in use
EADDRNOTAVAIL	socket address not available
EAFNOSUPPORT	unsupported socket addressing family
EAGAIN	resource temporarily unavailable
EALREADY	previous connection not yet completed
EARG	undefined function argument value
EBADF	file or socket not open or suitable (synonym for **ENOTOPEN**)
EBUSY	resource busy (synonym for **EINUSE**)
ECHILD	child process not found
ECONNABORTED	connection aborted by local network software
ECONNREFUSED	destination host refused socket connection
ECONNRESET	connection reset by peer
ECONV	data conversion failure
ECORRUPT	file is in a corrupt or unreadable state
EDEADLK	resource deadlock avoided
EDESTADRREQ	socket operation requires destination address
EDEVICE	physical device error
EDOM	math function domain error
EDUPKEY	attempt to add record with duplicate key
EEXIST	file already exists
EFATTR	file attribute conflict
EFAULT	invalid argument address
EFBIG	file too large
EFFORM	file format error
EFORBID	function execution prevented by run-time options
EHOSTDOWN	destination host is down
EHOSTUNREACH	destination host is unreachable

EILSEQ	error in multi-byte character sequence (reserved for future use)
EINPROGRESS	socket connection in progress
EINTR	function failed due to interruption by signal
EINUSE	file to be opened was already in use
EINVAL	invalid argument (synonym for **EARG**)
EIO	physical I/O error (synonym for **EDEVICE**)
EISCONN	socket is already connected
EISDIR	output file is a directory
ELIBERR	run-time system internal error
ELIMIT	internal limit exceeded
ELOOP	too many symbolic links in pathname (1003.1a)
EMAIN	synonymous error (synonym for **ESYN**)
EMFILE	too many open files (synonym for **ELIMIT**)
EMLINK	system limit on links exceeded
EMSGSIZE	message too large for datagram socket
EMVSCATLG	OpenEdition catalog OBTAIN error
EMVSCVAF	OpenEdition CVAF error
EMVSDYNALC	OpenEdition dynamic allocation error
EMVSERR	OpenEdition system error
EMVSEXPIRE	expired password
EMVSINITIAL	error in establishing OpenEdition process
EMVSNOTUP	OpenEdition kernel is not active
EMVSPASSWORD	incorrect password
EMVSPFSFILE	OpenEdition physical file error
EMVSPFSPERM	OpenEdition HFS system error
EMVSSSAF2ERR	OpenEdition security error
EMVSSAFEXTRERR	OpenEdition security extract error
ENAMETOOLONG	filename too long
ENETDOWN	local host's network down or inaccessible
ENETRESET	remote host dropped network communications
ENETUNREACH	destination network is unreachable
ENFILE	too many open HFS files in system
ENFOUND	file not found
ENOBUFS	insufficient buffers in network software
ENODEV	inappropriate use of device
ENOENT	file or directory not found (synonym for **ENFOUND**)
ENOEXEC	attempt to execute non-executable file
ENOLCK	no HFS record locks were available
ENOMEM	insufficient memory
ENOPROTOOPT	option not supported for protocol type
ENOSPC	no space in file

ENOSYS	function not implemented by system
ENOTCONN	socket is not connected
ENOTDIR	pathname component not a directory
ENOTEMPTY	directory not empty
ENOTOPEN	synonym for **EBADF**
ENOTSOCK	file descriptor not associated with a socket
ENOTTY	file is not a terminal
ENXIO	non-existent or inappropriate device
EOPNOTSUPP	operation not supported on socket
EPERM	operation not permitted
EPFNOSUPPORT	unsupported socket protocol family
EPIPE	write to pipe with no reader
EPREV	previous error not cleared
EPROTONOSUPPORT	unsupported socket protocol
EPROTOTYPE	protocol inconsistent with socket type
ERANGE	math function range error
EROFS	file system mounted read only
ESHUTDOWN	connection has been shut down
ESOCKTNOSUPPORT	socket type not allowed
ESPIPE	seek to unseekable file (synonym for **EUNSUPP**)
ESRCH	process not found
ESYS	operating system interface failure
ETIMEDOUT	socket connection attempt timed out
EUNSUPP	unsupported I/O operation
EUSAGE	incorrect function usage.
EWOULDBLOCK	socket operation would block
EXDEV	link from one file system to another

Your Turn

If you have comments or suggestions about *SAS/C Software Diagnostic Messages, Release 6.50*, please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Institute Inc.
Publications Division
SAS Campus Drive
Cary, NC 27513
email: yourturn@unx.sas.com

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: suggest@unx.sas.com