

Compaq Visual Fortran

Error Messages

Date: August, 2001

**Software
Version:** Visual Fortran Version 6.6

**Operating
Systems:** Microsoft Windows 98, Windows Me, Windows 95,
Windows 2000, or Windows NT Version 4

Compaq Computer Corporation
Houston, Texas

Copyright Information

Confidential computer software. Valid license from Compaq required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

© 2001 Compaq Computer Corporation

Visual Fortran Home Page, Photographic images: Copyright © 1997 PhotoDisc, Inc.

Visual Fortran Home Page, Image: CERN, European Laboratory for Particle Physics: ALICE detector on CERN's future accelerator, the LHC, Large Hadron Collider.

Compaq, the COMPAQ logo, DEC, DEC Fortran, DIGITAL, OpenVMS, Tru64, VAX, VAX FORTRAN, and VMS are trademarks of Compaq Information Technologies Group, L.P. in the United States and other countries.

ActiveX, Microsoft, Microsoft Press, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual J++, Visual Studio, Win32, Win32s, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States and other countries.

Intel and Pentium are trademarks of Intel Corporation in the United States and other countries.

AMD and Athlon are trademarks of Advanced Micro Devices, Inc.

Borland and Delphi are trademarks of Borland Software Corporation.

CRAY is a registered trademark of Cray Research, Inc.

IBM is a registered trademark of International Business Machines, Inc.

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers,

Inc.

IMSL and Visual Numerics are registered trademarks of Visual Numerics, Inc.

Linux is a registered trademark of Linus Torvalds.

OpenGL is a registered trademark of Silicon Graphics, Inc.

OpenMP and the OpenMP logo are trademarks of OpenMP Architecture Review Board.

Sun Microsystems is a registered trademark and Java is a trademark of Sun Microsystems, Inc.

UNIX is a trademark of the Open Group in the United States and other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Contents

In this document, links are denoted by a  or  when you pass your pointer over a blue-colored term. In either case, click on the link to see further information.

This section contains information on the following:

- [Visual Fortran Run-Time Errors](#)
- [Linker Tools Messages \(LNKxxxx\)](#)
- [BSCMAKE Messages \(BKxxxx\)](#)
- [C Run-Time Errors \(Rxxxx\)](#)
- [CVPACK Messages \(CKxxxx\)](#)
- [CVTRES Messages \(CVTxxxx\)](#)
- [Math Errors \(M6xxx\)](#)
- [NMAKE Messages \(Uxxxx\)](#)
- [Profiler Errors \(PRFxxxx\)](#)
- [Resource Compiler Messages \(RCxxxx\)](#)
- [Resource Compiler Messages \(RWxxxx\)](#)

Visual Fortran Run-Time Errors

This section describes Visual Fortran run-time error messages. For each error, the tables in the following sections provide the error number, the severity code, error message text, condition symbol name, and a detailed description of the errors.

The severity of the message determines whether program execution continues:

Severity	Description
<i>severe</i>	Must be corrected. The program's execution is terminated when the error is encountered, unless the program's I/O statements use the END, EOR, or ERR branch specifiers to transfer control, perhaps to a routine that uses the IOSTAT specifier.
<i>error</i>	Should be corrected. The program might continue execution, but the output from this execution may be incorrect.
<i>warning</i>	Should be investigated. The program continues execution, but output from this execution may be incorrect.
<i>info</i>	For informational purposes only; the program continues.

When a *severe* error occurs for which no recovery method (such as ERR, END, or EOR) is specified:

- The severe error message is displayed
- A hexadecimal dump of the call stack (program counter trace) is displayed
- Program execution stops

To prevent program termination of a severe error, consider including an I/O error-handling specifier (see [Methods of Handling Errors](#)).

For information about where a message is displayed (depends on the project type) and related information, see [Run-Time Message Display and Format](#).

The following sections describe the run-time errors in detail:

- [Run-Time Errors having no numbers and errors 1 through 30](#)
- [Run-Time Errors 31 through 60](#)
- [Run-Time Errors 61 through 108](#)
- [Run-Time Errors 120 through 300](#)

- [Run-Time Errors 540 through 569](#)
- [Run-Time Errors 570 through 614](#)
- [Run-Time Errors 615 through 656](#)
- [Run-Time Errors 657 through 689](#)
- [Run-Time Errors 690 through 719](#)
- [Run-Time Errors 720 through 759](#)

Related topics:

- [Default Run-Time Error Processing](#)
- [Handling Run-Time Errors](#)
- [Locating Run-Time Errors](#)
- [Run-Time Environment Variables](#)
- [The Floating-Point Environment](#)

Run-Time Errors Having No Numbers and Errors 1 Through 30

This section describes in detail Visual Fortran run-time errors having no numbers and run-time error numbers 1 through 30.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about the severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
None ¹	<p><i>info: Fortran error message number is nnn</i></p> <p>The Compaq Fortran message catalog file was not found on this system. This error has no condition symbol.</p>

None ¹	<p>warning: Could not open message catalog: for_msg.cat</p> <p>The Compaq Fortran message catalog file was not found on this system. This error has no condition symbol.</p>
None ¹	<p>Insufficient memory to open Fortran RTL catalog: message 41</p> <p>The Compaq Fortran message catalog file could not be opened because of insufficient virtual memory. For more information, see error 41. This error has no condition symbol.</p>
1 ¹	<p>severe (1): Not a Fortran-specific error</p> <p>FOR\$IOS_NOTFORSPE. An error in the user program or in the RTL was not a Compaq Fortran-specific error and was not reportable through any other Compaq Fortran run-time messages.</p>
8	<p>severe (8): Internal consistency check failure</p> <p>FOR\$IOS_BUG_CHECK. Internal error. Please check that the program is correct. Recompile if an error existed in the program. If this error persists, submit a problem report.</p>
9	<p>severe (9): Permission to access file denied</p> <p>FOR\$IOS_PERACCFIL. Check the permissions of the specified file and whether the network device is mapped and available. Make sure the correct file and device was being accessed. Change the protection, specific file, or process used before rerunning the program.</p>
10	<p>severe (10): Cannot overwrite existing file</p> <p>FOR\$IOS_CAVOVEEXI. Specified file <i>xxx</i> already exists when OPEN statement specified STATUS='NEW' (create new file) using I/O unit <i>x</i>. Make sure correct file name, directory path, unit, and so forth were specified in the source program. Decide whether to:</p> <ul style="list-style-type: none"> • Rename or remove the existing file before rerunning the program. • Modify the source file to specify different file specification, I/O unit, or OPEN statement STATUS.

11 ¹	<p><i>info (11): Unit not connected</i></p> <p>FOR\$IOS_UNINOTCON. The specified unit was not open at the time of the attempted I/O operation. Check if correct unit number was specified. If appropriate, use an OPEN statement to explicitly open the file (connect the file to the unit number).</p>
17	<p><i>severe (17): Syntax error in NAMELIST input</i></p> <p>FOR\$IOS_SYNERNAM. The syntax of input to a namelist-directed READ statement was incorrect.</p>
18	<p><i>severe (18): Too many values for NAMELIST variable</i></p> <p>FOR\$IOS_TOOMANVAL. An attempt was made to assign too many values to a variable during a namelist READ statement.</p>
19	<p><i>severe (19): Invalid reference to variable in NAMELIST input</i></p> <p>FOR\$IOS_INVREFVAR. One of the following conditions occurred:</p> <ul style="list-style-type: none"> • The variable was not a member of the namelist group. • An attempt was made to subscript a scalar variable. • A subscript of the array variable was out-of-bounds. • An array variable was specified with too many or too few subscripts for the variable. • An attempt was made to specify a substring of a noncharacter variable or array name. • A substring specifier of the character variable was out-of-bounds. • A subscript or substring specifier of the variable was not an integer constant. • An attempt was made to specify a substring by using an unsubscripted array variable.
20	<p><i>severe (20): REWIND error</i></p> <p>FOR\$IOS_REWERR. One of the following conditions occurred:</p> <ul style="list-style-type: none"> • The file was not a sequential file. • The file was not opened for sequential or append access. • The Compaq Fortran RTL I/O system detected an error condition during execution of a REWIND statement.

21	<p><i>severe (21): Duplicate file specifications</i></p> <p>FOR\$IOS_DUPFILSPE. Multiple attempts were made to specify file attributes without an intervening close operation. A DEFINE FILE statement was followed by another DEFINE FILE statement or an OPEN statement.</p>
22	<p><i>severe (22): Input record too long</i></p> <p>FOR\$IOS_INPRECTOO. A record was read that exceeded the explicit or default record length specified when the file was opened. To read the file, use an OPEN statement with a RECL= value (record length) of the appropriate size.</p>
23	<p><i>severe (23): BACKSPACE error</i></p> <p>FOR\$IOS_BACERR. The Compaq Fortran RTL I/O system detected an error condition during execution of a BACKSPACE statement.</p>
24 ¹	<p><i>severe (24): End-of-file during read</i></p> <p>FOR\$IOS_ENDDURREA. One of the following conditions occurred:</p> <ul style="list-style-type: none"> • A Compaq Fortran RTL I/O system end-of-file condition was encountered during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification. • An end-of-file record written by the ENDFILE statement was encountered during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification. • An attempt was made to read past the end of an internal file character string or array during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification. <p>This error is returned by END and ERRSNS.</p>
25	<p><i>severe (25): Record number outside range</i></p> <p>FOR\$IOS_RECNUMOUT. A direct access READ, WRITE, or FIND statement specified a record number outside the range specified when the file was opened.</p>

26	<p><i>severe (26): OPEN or DEFINE FILE required</i></p> <p>FOR\$IOS_OPEDEFREQ. A direct access READ, WRITE, or FIND statement was attempted for a file when no prior DEFINE FILE or OPEN statement with ACCESS='DIRECT' was performed for that file.</p>
27	<p><i>severe (27): Too many records in I/O statement</i></p> <p>FOR\$IOS_TOOMANREC. An attempt was made to do one of the following:</p> <ul style="list-style-type: none"> • Read or write more than one record with an ENCODE or DECODE statement. • Write more records than existed.
28	<p><i>severe (28): CLOSE error</i></p> <p>FOR\$IOS_CLOERR. An error condition was detected by the Compaq Fortran RTL I/O system during execution of a CLOSE statement.</p>
29	<p><i>severe (29): File not found</i></p> <p>FOR\$IOS_FILNOTFOU. A file with the specified name could not be found during an open operation.</p>
30	<p><i>severe (30): Open failure</i></p> <p>FOR\$IOS_OPEFAI. An error was detected by the Compaq Fortran RTL I/O system while attempting to open a file in an OPEN, INQUIRE, or other I/O statement. This message is issued when the error condition is not one of the more common conditions for which specific error messages are provided. It can occur when an OPEN operation was attempted for one of the following:</p> <ul style="list-style-type: none"> • Segmented file that was not on a disk or a raw magnetic tape • Standard I/O file that had been closed

¹Identifies errors not returned by IOSTAT.

Run-Time Errors 31 Through 60

This section describes in detail Visual Fortran run-time errors 31 through 60.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
31	<p><i>severe (31): Mixed file access modes</i></p> <p>FOR\$IOS_MIXFILACC. An attempt was made to use any of the following combinations:</p> <ul style="list-style-type: none"> • Formatted and unformatted operations on the same unit • An invalid combination of access modes on a unit, such as direct and sequential • A Compaq Fortran RTL I/O statement on a logical unit that was opened by a program coded in another language
32	<p><i>severe (32): Invalid logical unit number</i></p> <p>FOR\$IOS_INVLOGUNI. A logical unit number greater than 2,147,483,647 or less than zero was used in an I/O statement.</p>
33	<p><i>severe (33): ENDFILE error</i></p> <p>FOR\$IOS_ENDFILERR. One of the following conditions occurred:</p> <ul style="list-style-type: none"> • The file was not a sequential organization file with variable-length records. • The file was not opened for sequential, append, or direct access. • An unformatted file did not contain segmented records. • The Compaq Fortran RTL I/O system detected an error during execution of an ENDFILE statement.

34	<p><i>severe (34): Unit already open</i></p> <p>FOR\$IOS_UNIALROPE. A DEFINE FILE statement specified a logical unit that was already opened.</p>
35	<p><i>severe (35): Segmented record format error</i></p> <p>FOR\$IOS_SEGRECFOR. An invalid segmented record control data word was detected in an unformatted sequential file. The file was probably either created with RECORDTYPE='FIXED' or 'VARIABLE' in effect, or was created by a program written in a language other than Fortran or Fortran 90.</p>
36	<p><i>severe (36): Attempt to access non-existent record</i></p> <p>FOR\$IOS_ATTACCNON. A direct-access READ or FIND statement attempted to access beyond the end of a relative file (or a sequential file on disk with fixed-length records) or access a record that was previously deleted from a relative file.</p>
37	<p><i>severe (37): Inconsistent record length</i></p> <p>FOR\$IOS_INCRECLEN. An attempt was made to open a direct access file without specifying a record length.</p>
38	<p><i>severe (38): Error during write</i></p> <p>FOR\$IOS_ERRDURWRI. The Compaq Fortran RTL I/O system detected an error condition during execution of a WRITE statement.</p>
39	<p><i>severe (39): Error during read</i></p> <p>FOR\$IOS_ERRDURREA. The Compaq Fortran RTL I/O system detected an error condition during execution of a READ statement.</p>
40	<p><i>severe (40): Recursive I/O operation</i></p> <p>FOR\$IOS_RECIO_OPE. While processing an I/O statement for a logical unit, another I/O operation on the same logical unit was attempted, such as a function subprogram that performs I/O to the same logical unit that was referenced in an expression in an I/O list or variable format expression.</p>

41 ^{1,2}	<p><i>severe (41): Insufficient virtual memory</i></p> <p>FOR\$IOS_INSVIRMEM. The Compaq Fortran RTL attempted to exceed its available virtual memory while dynamically allocating space. To overcome this problem, investigate increasing the data limit (Control Panel, System, Performance Tab, Virtual Memory). Before you try to run this program again, wait until the new system resources take effect.</p>
42	<p><i>severe (42): No such device</i></p> <p>FOR\$IOS_NO_SUCDEV. A pathname included an invalid or unknown device name when an OPEN operation was attempted.</p>
43	<p><i>severe (43): File name specification error</i></p> <p>FOR\$IOS_FILNAM_SPE. A pathname or file name given to an OPEN or INQUIRE statement was not acceptable to the Compaq Fortran RTL I/O system.</p>
44	<p><i>severe (44): Inconsistent record type</i></p> <p>FOR\$IOS_INCRECTYP. The RECORDTYPE value in an OPEN statement did not match the record type attribute of the existing file that was opened.</p>
45	<p><i>severe (45): Keyword value error in OPEN statement</i></p> <p>FOR\$IOS_KEYVALERR. An improper value was specified for an OPEN or CLOSE statement specifier requiring a value.</p>
46	<p><i>severe (46): Inconsistent OPEN/CLOSE parameters</i></p> <p>FOR\$IOS_INCOPECLO. Specifications in an OPEN or CLOSE statement were inconsistent. Some invalid combinations follow:</p> <ul style="list-style-type: none"> • READONLY or ACTION='READ' with STATUS='NEW' or STATUS='SCRATCH' • READONLY with STATUS='REPLACE', ACTION='WRITE', or ACTION='READWRITE' • ACCESS='APPEND' with READONLY, ACTION='READ', STATUS='NEW', or STATUS='SCRATCH' • DISPOSE='SAVE', 'PRINT', or 'SUBMIT' with STATUS='SCRATCH' • DISPOSE='DELETE' with READONLY

	<ul style="list-style-type: none"> • CLOSE statement STATUS='DELETE' with OPEN statement READONLY • ACCESS='APPEND' with STATUS='REPLACE' • ACCESS='DIRECT' with POSITION='APPEND' or 'ASIS'
47	<p><i>severe (47): Write to READONLY file</i></p> <p>FOR\$IOS_WRIREFIL. A write operation was attempted to a file that was declared ACTION='READ' or READONLY in the OPEN statement that is currently in effect.</p>
48	<p><i>severe (48): Invalid argument to Fortran Run-Time Library</i></p> <p>FOR\$IOS_INVARGFOR. The compiler passed an invalid or improperly coded argument to the Compaq Fortran RTL. This can occur if the compiler is newer than the RTL in use.</p>
51	<p><i>severe (51): Inconsistent file organization</i></p> <p>FOR\$IOS_INCFILORG. The file organization specified in an OPEN statement did not match the organization of the existing file.</p>
53	<p><i>severe (53): No current record</i></p> <p>FOR\$IOS_NO_CURREC. Attempted to execute a REWRITE statement to rewrite a record when the current record was undefined. To define the current record, execute a successful READ statement. You can optionally perform an INQUIRE statement on the logical unit after the READ statement and before the REWRITE statement. No other operations on the logical unit may be performed between the READ and REWRITE statements.</p>
55	<p><i>severe (55): DELETE error</i></p> <p>FOR\$IOS_DELERR. An error condition was detected by the Compaq Fortran RTL I/O system during execution of a DELETE statement.</p>
57	<p><i>severe (57): FIND error</i></p> <p>FOR\$IOS_FINERR. The Compaq Fortran RTL I/O system detected an error condition during execution of a FIND statement.</p>

58 ³	<p><i>info (58): Format syntax error at or near xx</i></p> <p>FOR\$IOS_FMTSYN. Check the statement containing <i>xx</i>, a character substring from the format string, for a format syntax error. For more information, see the FORMAT statement.</p>
59 ⁴	<p><i>severe (59): List-directed I/O syntax error</i></p> <p>FOR\$IOS_LISIO_SYN. The data in a list-directed input record had an invalid format, or the type of the constant was incompatible with the corresponding variable. The value of the variable was unchanged.</p>
60	<p><i>severe (60): Infinite format loop</i></p> <p>FOR\$IOS_INFFORLOO. The format associated with an I/O statement that included an I/O list had no field descriptors to use in transferring those values.</p>

¹Identifies errors that can be returned by STAT in an **ALLOCATE** statement.

²Identifies errors that can be returned by STAT in a **DEALLOCATE** statement.

³Identifies errors not returned by IOSTAT.

⁴The ERR transfer is taken after completion of the I/O statement for error number 59. The resulting file status and record position are the same as if no error had occurred. However, other I/O errors take the ERR transfer as soon as the error is detected, so file status and record position are undefined.

Run-Time Errors 61 Through 108

This section describes in detail Visual Fortran run-time errors 61 through 108.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about the severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
61	<p><i>severe or info</i>¹ (61): <i>Format/variable-type mismatch</i></p> <p>FOR\$IOS_FORVARMIS². An attempt was made either to read or write a real variable with an integer field descriptor (I, L, O, Z, B), or to read or write an integer or logical variable with a real field descriptor (D, E, or F). To suppress this error message, see the description of /check:noformat.</p>
62	<p><i>severe</i> (62): <i>Syntax error in format</i></p> <p>FOR\$IOS_SYNERFOR. A syntax error was encountered while the RTL was processing a format stored in an array or character variable.</p>
63	<p><i>error or info</i>¹ (63): <i>Output conversion error</i></p> <p>FOR\$IOS_OUTCONERR². During a formatted output operation, the value of a particular number could not be output in the specified field length without loss of significant digits. When this situation is encountered, the overflowed field is filled with asterisks to indicate the error in the output record. If no ERR address has been defined for this error, the program continues after the error message is displayed. To suppress this error message, see the description of /check:nooutput_conversion.</p>
64	<p><i>severe</i> (64): <i>Input conversion error</i></p> <p>FOR\$IOS_INPCONERR². During a formatted input operation, an invalid character was detected in an input field, or the input value overflowed the range representable in the input variable. The value of the variable was set to zero.</p>
65	<p><i>error</i> (65): <i>Floating invalid</i></p> <p>FOR\$IOS_FLTINV. During an arithmetic operation, the floating-point values used in a calculation were invalid for the type of operation requested or invalid exceptional values. For example, the error can occur if you request a log of the floating-point values 0.0 or a negative number. For certain arithmetic expressions, specifying the /check:nopower option can suppress this message.</p>

66	<p><i>severe (66): Output statement overflows record</i></p> <p>FOR\$IOS_OUTSTAOVE. An output statement attempted to transfer more data than would fit in the maximum record size.</p>
67	<p><i>severe (67): Input statement requires too much data</i></p> <p>FOR\$IOS_INPSTAREQ. Attempted to read more data than exists in a record with an unformatted READ statement or with a formatted sequential READ statement from a file opened with a PAD specifier value of 'NO'.</p>
68	<p><i>severe (68): Variable format expression value error</i></p> <p>FOR\$IOS_VFEVALERR². The value of a variable format expression was not within the range acceptable for its intended use; for example, a field width was less than or equal to zero. A value of 1 was assumed, except for a P edit descriptor, for which a value of zero was assumed.</p>
70 ³	<p><i>severe (70): Integer overflow</i></p> <p>FOR\$IOS_INTOVF. During an arithmetic operation, an integer value exceeded byte, word, or longword range. The result of the operation was the correct low-order part. See the Data Representation chapter in the <i>Programmer's Guide</i> for ranges of the various integer data types. Consider specifying a larger integer data size (modify source program or, for an INTEGER declaration, possibly use the /integer size:size option).</p>
71 ³	<p><i>severe (71): Integer divide by zero</i></p> <p>FOR\$IOS_INTDIV. During an integer arithmetic operation, an attempt was made to divide by zero. The result of the operation was set to the dividend, which is equivalent to division by 1.</p>
72 ³	<p><i>error (72): Floating overflow</i></p> <p>FOR\$IOS_FLTOVF. During an arithmetic operation, a floating-point value exceeded the largest representable value for that data type. See the Data Representation chapter in the <i>Programmer's Guide</i> for ranges of the various data types.</p>

73 ³	<p><i>error (73): Floating divide by zero</i></p> <p>FOR\$IOS_FLTDIV. During a floating-point arithmetic operation, an attempt was made to divide by zero.</p>
74 ³	<p><i>error (74): Floating underflow</i></p> <p>FOR\$IOS_FLTUND. During an arithmetic operation, a floating-point value became less than the smallest finite value for that data type. Depending on the values of the /fpe:n option, the underflowed result was either set to zero or allowed to gradually underflow. See the Data Representation chapter in the <i>Programmer's Guide</i> for ranges of the various data types.</p>
75 ³	<p><i>error (75): Floating point exception</i></p> <p>FOR\$IOS_SIGFPE. A floating-point exception occurred. Possible causes include:</p> <ul style="list-style-type: none"> • Division by zero • Overflow • An invalid operation, such as subtraction of infinite values, multiplication of zero by infinity without signs), division of zero by zero or infinity by infinity • Conversion of floating-point to fixed-point format when an overflow prevents conversion
77 ³	<p><i>severe (77): Subscript out of range</i></p> <p>FOR\$IOS_SUBRNG. An array reference was detected outside the declared array bounds.</p>
78 ³	<p><i>error (78): Process killed</i></p> <p>FOR\$IOS_SIGTERM. The process received a signal requesting termination of this process. Determine the source of this software termination signal.</p>
79 ³	<p><i>error (79): Process quit</i></p> <p>FOR\$IOS_SIGQUIT. The process received a signal requesting termination of itself. Determine the source of this quit signal.</p>

95 ³	<p><i>info (95): Floating-point conversion failed</i></p> <p>FOR\$IOS_FLOCONFAI. The attempted unformatted read or write of nonnative floating-point data failed because the floating-point value:</p> <ul style="list-style-type: none"> • Exceeded the allowable maximum value for the equivalent native format and was set equal to infinity (plus or minus) • Was infinity (plus or minus) and was set to infinity (plus or minus) • Was invalid and was set to not a number (NaN) <p>Very small numbers are set to zero (0). This error could be caused by the specified nonnative floating-point format not matching the floating-point format found in the specified file. Check the following:</p> <ul style="list-style-type: none"> • The correct file was specified. • The record layout matches the format Compaq Fortran is expecting. • The ranges for the data being used (see Data Representation). • The correct nonnative floating-point data format was specified (see Converting Unformatted Numeric Data).
108	<p><i>Severe (108): Cannot stat file</i></p> <p>FOR\$IOS_CANSTAFI. Make sure correct file and unit were specified.</p>

¹For errors 61 and 63, the severity depends on the [/check:keywords](#) used during compilation command.

²The ERR transfer is taken after completion of the I/O statement for error numbers 61, 63, 64, and 68. The resulting file status and record position are the same as if no error had occurred. However, other I/O errors take the ERR transfer as soon as the error is detected, so file status and record position are undefined.

³Identifies errors not returned by IOSTAT.

Run-Time Errors 120 Through 300

This section describes in detail Visual Fortran run-time errors 120 through 300.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about the severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
120	<p><i>severe (120): Operation requires seek ability</i></p> <p>FOR\$IOS_OPEREQSEE. Attempted an operation on a file that requires the ability to perform seek operations on that file. Make sure the correct unit, directory path, and file were specified.</p>
138 ¹	<p><i>severe (138): Array index out of bounds</i></p> <p>FOR\$IOS_BRK_RANGE. An array subscript is outside the dimensioned boundaries of that array. Recompile with the /check:bounds option set.</p>
139 ¹	<p><i>severe: (139): Array index out of bounds for index nn</i></p> <p>FOR\$IOS_BRK_RANGE2. An array subscript is outside the dimensioned boundaries of that array. Recompile with the /check:bounds option set.</p>
140 ¹	<p><i>error (140): Floating inexact</i></p> <p>FOR\$IOS_FLTINE. A floating-point arithmetic or conversion operation gave a result that differs from the mathematically exact result. This trap is reported if the rounded result of an IEEE operation is not exact.</p>
144 ¹	<p><i>severe (144): Reserved operand</i></p> <p>FOR\$IOS_ROPRAND. The Compaq Fortran RTL encountered a reserved operand while executing your program. Please report the problem to Compaq.</p>

145 ¹	<p><i>severe (145): Assertion error</i></p> <p>FOR\$IOS_ASSERTERR. The Compaq Fortran RTL encountered an assertion error. Please report the problem to Compaq.</p>
146 ¹	<p><i>severe (146): Null pointer error</i></p> <p>FOR\$IOS_NULPTRERR. Attempted to use a pointer that does not contain an address. Modify the source program, recompile, and relink.</p>
147 ¹	<p><i>severe (147): Stack overflow</i></p> <p>FOR\$IOS_STKOVF. The Compaq Fortran RTL encountered a stack overflow while executing your program. See the Section "Guidelines for Programs That Use Large Data Arrays" in the post-installation part of <i>Compaq Visual Fortran Installing and Getting Started</i>.</p>
148 ¹	<p><i>severe (148): String length error</i></p> <p>FOR\$IOS_STRLENERR. During a string operation, an integer value appears in a context where the value of the integer is outside the permissible string length range. Recompile with the /check:bounds option.</p>
149 ¹	<p><i>severe (149): Substring error</i></p> <p>FOR\$IOS_SUBSTRERR. An array subscript is outside the dimensioned boundaries of an array. Recompile with the /check:bounds option.</p>
150 ¹	<p><i>severe (150): Range error</i></p> <p>FOR\$IOS_RANGEERR. An integer value appears in a context where the value of the integer is outside the permissible range.</p>
151 ^{1,2}	<p><i>severe (151): Allocatable array is already allocated</i></p> <p>FOR\$IOS_INVREALLOC. An allocatable array must not already be allocated when you attempt to allocate it. You must deallocate the array before it can again be allocated.</p>

152 ¹	<p><i>severe (152): Unresolved contention for DEC Fortran RTL global resource</i></p> <p>FOR\$IOS_RESACQFAI. Failed to acquire a Compaq Fortran RTL global resource for a reentrant routine. For a multithreaded program, the requested global resource is held by a different thread in your program. For a program using asynchronous handlers, the requested global resource is held by the calling part of the program (such as main program) and your asynchronous handler attempted to acquire the same global resource.</p>
153 ^{1,3}	<p><i>severe (153): Allocatable array or pointer is not allocated</i></p> <p>FOR\$IOS_INVDEALLOC. A Fortran 90 allocatable array or pointer must already be allocated when you attempt to deallocate it. You must allocate the array or pointer before it can again be deallocated.</p>
154 ¹	<p><i>severe(154): Array index out of bounds</i></p> <p>FOR\$IOS_RANGE. An array subscript is outside the dimensioned boundaries of that array. Recompile with the /check:bounds option set.</p>
155 ¹	<p><i>severe(155): Array index out of bounds for index nn</i></p> <p>FOR\$IOS_RANGE2. An array subscript is outside the dimensioned boundaries of that array. Recompile with the /check:bounds option set.</p>
156 ¹	<p><i>severe(156): GENTRAP code = hex dec</i></p> <p>FOR\$IOS_DEF_GENTRAP. The Compaq Fortran RTL has detected an unknown GENTRAP code. The cause is most likely a software problem due to memory corruption, or software signalling an exception with an incorrect exception code. Try recompiling with the /check:bounds option set to see if that finds the problem.</p>
157 ¹	<p><i>severe(157): Program Exception - access violation</i></p> <p>FOR\$IOS_ACCVIO. The program tried to read from or write to a virtual address for which it does not have the appropriate access. Try recompiling with the /check:bounds and /warn:argument checking options set, to see if the problem is an out-of-bounds memory reference or a argument mismatch that causes data to be treated as an address.</p> <p>Other causes of this error include:</p>

	<ul style="list-style-type: none"> • Mismatches in C vs. STDCALL calling mechanisms, causing the stack to become corrupted • References to unallocated pointers • Attempting to access a protected (for example, read-only) address
158 ¹	<p><i>severe(158): Program Exception - datatype misalignment</i></p> <p>FOR\$IOS_DTYPE_MISALIGN. The Compaq Fortran RTL has detected data that is not aligned on a natural boundary for the data type specified. For example, a REAL(8) data item aligned on natural boundaries has an address that is a multiple of 8. To ensure naturally aligned data, use the /alignment option. For more information on alignment, see Data Alignment Considerations in the <i>Programmer's Guide</i>.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
159 ¹	<p><i>severe(159): Program Exception - breakpoint</i></p> <p>FOR\$IOS_PGM_BPT. The Compaq Fortran RTL has encountered a breakpoint in the program.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
160 ¹	<p><i>severe(160): Program Exception - single step</i></p> <p>FOR\$IOS_PGM_SS. A trace trap or other single-instruction mechanism has signaled that one instruction has been executed.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
161 ¹	<p><i>severe(161): Program Exception - array bounds exceeded</i></p> <p>FOR\$IOS_PGM_BOUNDS. The program tried to access an array element that is outside the specified boundaries of the array. Recompile with the /check:bounds option set.</p>

162 ¹	<p><i>severe(162): Program Exception - denormal floating-point operand</i></p> <p>FOR\$IOS_PGM_DENORM. A floating-point arithmetic or conversion operation has a denormalized number as an operand. A denormalized number is smaller than the lowest value in the normal range for the data type specified. See the Data Representation chapter in the <i>Programmer's Guide</i> for ranges for floating-point types.</p> <p>Either locate and correct the source code causing the denormalized value or, if a denormalized value is acceptable, specify a different value for the /fpe compiler option to allow program continuation.</p>
163 ¹	<p><i>severe(163): Program Exception - floating stack check</i></p> <p>FOR\$IOS_PGM_FLTSTK. During a floating-point operation, the floating-point register stack on ia32 systems overflowed or underflowed. This is a fatal exception. The most likely cause is calling a REAL function as if it were an INTEGER function or subroutine, or calling an INTEGER function or subroutine as if it were a REAL function.</p> <p>Carefully check that the calling code and routine being called agree as to how the routine is declared. If you are unable to resolve the issue, please send a problem report with an example to Compaq.</p>
164 ¹	<p><i>severe(164): Program Exception - integer divide by zero</i></p> <p>FOR\$IOS_PGM_INTDIV. During an integer arithmetic operation, an attempt was made to divide by zero. Locate and correct the source code causing the integer divide by zero.</p>
165 ¹	<p><i>severe(165): Program Exception - integer overflow</i></p> <p>FOR\$IOS_PGM_INTOVF. During an arithmetic operation, an integer value exceeded the largest representable value for that data type. See the Data Representation chapter in the <i>Programmer's Guide</i> for ranges for INTEGER types.</p> <p>This message appears when /check:overflow was specified and an integer overflow appears.</p>

166 ¹	<p><i>severe(166): Program Exception - privileged instruction</i></p> <p>FOR\$IOS_PGM_PRIVINST. The program tried to execute an instruction whose operation is not allowed in the current machine mode.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
167 ¹	<p><i>severe(167): Program Exception - in page error</i></p> <p>FOR\$IOS_PGM_INPGERR. The program tried to access a page that was not present, so the system was unable to load the page. For example, this error might occur if a network connection was lost while trying to run a program over the network.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
168 ¹	<p><i>severe(168): Program Exception - illegal instruction</i></p> <p>FOR\$IOS_PGM_ILLINST. The program tried to execute an invalid instruction.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
169 ¹	<p><i>severe(169): Program Exception - noncontinuable exception</i></p> <p>FOR\$IOS_PGM_NOCONTEXCP. The program tried to continue execution after a noncontinuable exception occurred.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
170 ¹	<p><i>severe(170): Program Exception - stack overflow</i></p> <p>FOR\$IOS_PGM_STKOVF. The Compaq Fortran RTL has detected a stack overflow while executing your program. See your Release Notes for information on how to increase stack size.</p>

171 ¹	<p><i>severe(171): Program Exception - invalid disposition</i></p> <p>FOR\$IOS_PGM_INVDISP. An exception handler returned an invalid disposition to the exception dispatcher. Programmers using a high-level language should never encounter this exception.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
172 ¹	<p><i>severe(172): Program Exception - exception code = hex dec</i></p> <p>FOR\$IOS_PGM_EXCP_CODE. The Compaq Fortran RTL has detected an unknown exception code.</p> <p>This is an operating system error. See your operating system documentation for more information.</p>
173 ^{1,3}	<p><i>severe(173): A pointer passed to DEALLOCATE points to an array that cannot be deallocated</i></p> <p>FOR\$IOS_INVDEALLOC2. A pointer that was passed to DEALLOCATE pointed to an explicit array, an array slice, or some other type of memory that could not be deallocated in a DEALLOCATE statement. Only whole arrays previous allocated with an ALLOCATE statement may be validly passed to DEALLOCATE.</p>
175 ¹	<p><i>severe(175): DATE argument to DATE_AND_TIME is too short (LEN=n), required LEN=8</i></p> <p>FOR\$IOS_SHORTDATEARG. The number of characters associated with the DATE argument to the DATE AND TIME intrinsic was shorter than the required length. You must increase the number of characters passed in for this argument to be at least 8 characters in length. Verify that the TIME and ZONE arguments also meet their minimum lengths.</p>
176 ¹	<p><i>severe(176): TIME argument to DATE_AND_TIME is too short (LEN=n), required LEN=10</i></p> <p>FOR\$IOS_SHORTTIMEARG. The number of characters associated with the TIME argument to the DATE AND TIME intrinsic was shorter than the required length. You must increase the number of characters passed in for this argument to be at least 10 characters in length. Verify that the DATE and ZONE arguments also meet their</p>

	minimum lengths.
177 ¹	<p><i>severe(177): ZONE argument to DATE_AND_TIME is too short (LEN=n), required LEN=5</i></p> <p>FOR\$IOS_SHORTZONEARG. The number of characters associated with the ZONE argument to the DATE_AND_TIME intrinsic was shorter than the required length. You must increase the number of characters passed in for this argument to be at least 5 characters in length. Verify that the DATE and TIME arguments also meet their minimum lengths.</p>
178 ¹	<p><i>severe(178): Divide by zero</i></p> <p>FOR\$IOS_DIV. A floating-point or integer divide-by-zero exception occurred.</p>
179 ^{1,2}	<p><i>severe(179): Cannot allocate array - overflow on array size calculation</i></p> <p>FOR\$IOS_ARRSIZEOVF. An attempt to dynamically allocate storage for an array failed because the required storage size exceeds addressable memory.</p>
201 ¹	<p><i>severe (201): Intel Pentium fdiv flaw detected, please update the processor</i></p> <p>FOR\$IOS_FLAWEDPENT. The Fortran run-time system detected a flawed Pentium® processor. Please contact Intel® to upgrade your processor. To allow program execution to continue when this condition occurs, set the environment variable FOR_RUN_FLAWED_PENTIUM to true and rerun the program. For more information, see Intel Pentium Floating-Point Flaw.</p>
256	<p><i>severe (256): Unformatted I/O to unit open for formatted transfers</i></p> <p>FOR\$IOS_UNFIO_FMT. Attempted unformatted I/O to a unit where the OPEN statement (FORM specifier) indicated the file was formatted. Check that the correct unit (file) was specified. If the FORM specifier was not present in the OPEN statement and the file contains unformatted data, specify FORM='UNFORMATTED' in the OPEN statement. Otherwise, if appropriate, use formatted I/O (such as list-directed or namelist I/O).</p>

257	<p><i>severe (257): Formatted I/O to unit open for unformatted transfers</i></p> <p>FOR\$IOS_FMTIO_UNF. Attempted formatted I/O (such as list-directed or namelist I/O) to a unit where the OPEN statement indicated the file was unformatted (FORM specifier). Check that the correct unit (file) was specified. If the FORM specifier was not present in the OPEN statement and the file contains formatted data, specify FORM='FORMATTED' in the OPEN statement. Otherwise, if appropriate, use unformatted I/O.</p>
259	<p><i>severe (259): Sequential-access I/O to unit open for direct access</i></p> <p>FOR\$IOS_SEQIO_DIR. The OPEN for this unit number specified direct access and the I/O statement specifies sequential access. Check the OPEN statement and make sure the I/O statement uses the correct unit number and type of access.</p> <p>For more information on OPEN and other statements, see the A to Z Reference in the online Language Reference or the printed <i>Compaq Fortran Language Reference Manual</i>.</p>
264	<p><i>severe (264): operation requires file to be on disk or tape</i></p> <p>FOR\$IOS_OPEREQDIS. Attempted to use a BACKSPACE statement on such devices as a terminal.</p>
265	<p><i>severe (265): operation requires sequential file organization and access</i></p> <p>FOR\$IOS_OPEREQSEQ. Attempted to use a BACKSPACE statement on a file whose organization was not sequential or whose access was not sequential. A BACKSPACE statement can only be used for sequential files opened for sequential access.</p>
266 ¹	<p><i>error (266): Fortran abort routine called</i></p> <p>FOR\$IOS_PROABOUSE. The program called the abort routine to terminate itself.</p>
268 ¹	<p><i>severe (268): End of record during read</i></p> <p>FOR\$IOS_ENDRECDUR. An end-of-record condition was encountered during execution of a nonadvancing I/O READ statement that did not specify the EOR branch specifier.</p>

296 ¹	<i>info(296): nn floating inexact traps</i> FOR\$IOS_FLOINEEXC. The total number of floating-point inexact data traps encountered during program execution was <i>nn</i> . This summary message appears at program completion.
297 ¹	<i>info (297): nn floating invalid traps</i> FOR\$IOS_FLOINVEXC. The total number of floating-point invalid data traps encountered during program execution was <i>nn</i> . This summary message appears at program completion.
298 ¹	<i>info (298): nn floating overflow traps</i> FOR\$IOS_FLOOVFEXC. The total number of floating-point overflow traps encountered during program execution was <i>nn</i> . This summary message appears at program completion.
299 ¹	<i>info (299): nn floating divide-by-zero traps</i> FOR\$IOS_FLODIV0EXC. The total number of floating-point divide-by-zero traps encountered during program execution was <i>nn</i> . This summary message appears at program completion.
300 ¹	<i>info (300): nn floating underflow traps</i> FOR\$IOS_FLOUNDEXC. The total number of floating-point underflow traps encountered during program execution was <i>nn</i> . This summary message appears at program completion.

¹Identifies errors not returned by IOSTAT.

²Identifies errors that can be returned by STAT in an **ALLOCATE** statement.

³Identifies errors that can be returned by STAT in a **DEALLOCATE** statement.

Run-Time Errors 540 Through 569

This section describes in detail Visual Fortran run-time errors 540 through 569.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
540	<p><i>severe (540): Array or substring subscript expression out of range</i></p> <p>FOR\$IOS_F6096. An expression used to index an array was smaller than the lower dimension bound or larger than the upper dimension bound.</p>
541	<p><i>severe (541): CHARACTER substring expression out of range</i></p> <p>FOR\$IOS_F6097. An expression used to index a character substring was illegal.</p>
542	<p><i>severe (542): Label not found in assigned GOTO list</i></p> <p>FOR\$IOS_F6098. The label assigned to the integer-variable name was not specified in the label list of the assigned GOTO statement.</p>
543	<p><i>severe (543): INTEGER arithmetic overflow</i></p> <p>FOR\$IOS_F6099. This error occurs whenever integer arithmetic results in overflow.</p>
544	<p><i>severe (544): INTEGER overflow on input</i></p> <p>FOR\$IOS_F6100. An integer item exceeded the legal size limits.</p> <p>An INTEGER(1) item must be in the range -127 to 128. An INTEGER(2) item must be in the range -32,767 to 32,768. An INTEGER(4) item must be in the range -2,147,483,647 to 2,147,483,648.</p>

545	<p><i>severe (545): Invalid INTEGER</i></p> <p>FOR\$IOS_F6101. Either an illegal character appeared as part of an integer, or a numeric character larger than the radix was used in an alternate radix specifier.</p>
546	<p><i>severe (546): REAL indefinite (uninitialized or previous error)</i></p> <p>FOR\$IOS_F6102. An invalid real number was read from a file, an internal variable, or the console. This can happen if an invalid number is generated by passing an illegal argument to an intrinsic function -- for example, SQRT(-1) or ASIN(2). If the invalid result is written and then later read, the error will be generated.</p>
547	<p><i>severe (547): Invalid REAL</i></p> <p>FOR\$IOS_F103. An illegal character appeared as part of a real number.</p>
548	<p><i>severe (548): REAL math overflow</i></p> <p>FOR\$IOS_F6104. A real value was too large. Floating-point overflows in either direct or emulated mode generate NaN (Not-A-Number) exceptions, which appear in the output field as asterisks (*) or the letters NAN.</p>
550	<p><i>severe (550): INTEGER assignment overflow</i></p> <p>FOR\$IOS_F6106. This error occurs when assignment to an integer is out of range. This message appears only if the /check:bounds option is used in compiling.</p>
551	<p><i>severe (551): Formatted I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6200. The program tried to perform formatted I/O on a unit opened with FORM='UNFORMATTED' or FORM='BINARY'.</p>
552	<p><i>severe (552): List-directed I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6201. The program tried to perform list-directed I/O on a file that was not opened with FORM='FORMATTED' and ACCESS='SEQUENTIAL'.</p>

553	<p><i>severe (553): Terminal I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6202. When a special device such as CON, LPT1, or PRN is opened in an OPEN statement, its access must be sequential and its format must be either formatted or binary. By default ACCESS='SEQUENTIAL' and FORM='FORMATTED' in OPEN statements.</p> <p>To generate this error the device's OPEN statement must contain an option not appropriate for a terminal device, such as ACCESS='DIRECT' or FORM='UNFORMATTED'.</p>
554	<p><i>severe (554): Direct I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6203. A REC= option was included in a statement that transferred data to a file that was opened with the ACCESS='SEQUENTIAL' option.</p>
555	<p><i>severe (555): Unformatted I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6204. If a file is opened with FORM='FORMATTED', unformatted or binary data transfer is prohibited.</p>
556	<p><i>severe (556): A edit descriptor expected for CHARACTER</i></p> <p>FOR\$IOS_F6205. The A edit descriptor was not specified when a character data item was read or written using formatted I/O.</p>
557	<p><i>severe (557): E, F, D, or G edit descriptor expected for REAL</i></p> <p>FOR\$IOS_F6206. The E, F, D, or G edit descriptor was not specified when a real data item was read or written using formatted I/O.</p>
558	<p><i>severe (558): I edit descriptor expected for INTEGER</i></p> <p>FOR\$IOS_F6207. The I edit descriptor was not specified when an integer data item was read or written using formatted I/O.</p>
559	<p><i>severe (559): L edit descriptor expected for LOGICAL</i></p> <p>FOR\$IOS_F6208. The L edit descriptor was not specified when a logical data item was read or written using formatted I/O.</p>

560	<p><i>severe (560): File already open: parameter mismatch</i></p> <p>FOR\$IOS_F6209. An OPEN statement specified a connection between a unit and a filename that was already in effect. In this case, only the BLANK= option can have a different setting.</p>
561	<p><i>severe (561): Namelist I/O not consistent with OPEN options</i></p> <p>FOR\$IOS_F6210. The program tried to perform namelist I/O on a file that was not opened with FORM='FORMATTED' and ACCESS='SEQUENTIAL'.</p>
562	<p><i>severe (562): IOFOCUS option illegal with non-window unit</i></p> <p>FOR\$IOS_F6211. IOFOCUS was specified in an OPEN or INQUIRE statement for a non-window unit. The IOFOCUS option can only be used when the unit opened or inquired about is a QuickWin child window.</p>
563	<p><i>severe (563): IOFOCUS option illegal without QuickWin</i></p> <p>FOR\$IOS_F6212. IOFOCUS was specified in an OPEN or INQUIRE statement for a non-QuickWin application. The IOFOCUS option can only be used when the unit opened or inquired about is a QuickWin child window.</p>
564	<p><i>severe (564): TITLE illegal with non-window unit</i></p> <p>FOR\$IOS_F6213. TITLE was specified in an OPEN or INQUIRE statement for a non-window unit. The TITLE option can only be used when the unit opened or inquired about is a QuickWin child window.</p>
565	<p><i>severe (565): TITLE illegal without QuickWin</i></p> <p>FOR\$IOS_F6214. TITLE was specified in an OPEN or INQUIRE statement for a non-QuickWin application. The TITLE option can only be used when the unit opened or inquired about is a QuickWin child window.</p>
566	<p><i>severe (566): KEEP illegal for scratch file</i></p> <p>FOR\$IOS_F6300. STATUS='KEEP' was specified for a scratch file; this is illegal because scratch files are automatically deleted at program termination.</p>

567	<p><i>severe (567): SCRATCH illegal for named file</i></p> <p>FOR\$IOS_F6301. STATUS='SCRATCH' should not be used in a statement that includes a filename.</p>
568	<p><i>severe (568): Multiple radix specifiers</i></p> <p>FOR\$IOS_F6302. More than one alternate radix for numeric I/O was specified. F6302 can indicate an error in spacing or a mismatched format for data of different radices.</p>
569	<p><i>severe (569): Illegal radix specifier</i></p> <p>FOR\$IOS_F6303. A radix specifier was not between 2 and 36, inclusive. Alternate radix constants must be of the form n#ddd... where n is a radix from 2 to 36 inclusive and ddd... are digits with values less than the radix. For example, 3#12 and 34#7AX are valid constants with valid radix specifiers. 245#7A and 39#12 do not have valid radix specifiers and generate error 569 if input.</p>

Run-Time Errors 570 Through 614

This section describes in detail Visual Fortran run-time errors 570 through 614.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
--------	--

570	<p><i>severe (570): Illegal STATUS value</i></p> <p>FOR\$IOS_F6304. An illegal value was used with the STATUS option.</p> <p>STATUS accepts the following values:</p> <ul style="list-style-type: none"> • 'KEEP' or 'DELETE' when used with CLOSE statements • 'OLD', 'NEW', 'SCRATCH', or 'UNKNOWN' when used with OPEN statements
571	<p><i>severe (571): Illegal MODE value</i></p> <p>FOR\$IOS_F6305. An illegal value was used with the MODE option.</p> <p>MODE accepts the values 'READ', 'WRITE', or 'READWRITE'.</p>
572	<p><i>severe (572): Illegal ACCESS value</i></p> <p>FOR\$IOS_F6306. An illegal value was used with the ACCESS option.</p> <p>ACCESS accepts the values 'SEQUENTIAL' and 'DIRECT'.</p>
573	<p><i>severe (573): Illegal BLANK value</i></p> <p>FOR\$IOS_F6307. An illegal value was used with the BLANK option.</p> <p>BLANK accepts the values 'NULL' and 'ZERO'.</p>
574	<p><i>severe (574): Illegal FORM value</i></p> <p>FOR\$IOS_F6308. An illegal value was used with the FORM option.</p> <p>FORM accepts the following values: 'FORMATTED', 'UNFORMATTED', and 'BINARY'.</p>
575	<p><i>severe (575): Illegal SHARE value</i></p> <p>FOR\$IOS_F6309. An illegal value was used with the SHARE option.</p> <p>SHARE accepts the values 'COMPAT', 'DENYRW', 'DENYWR', 'DENYRD', and 'DENYNONE'.</p>

577	<p><i>severe (577): Illegal record number</i></p> <p>FOR\$IOS_F6311. An invalid number was specified as the record number for a direct-access file.</p> <p>The first valid record number for direct-access files is 1.</p>
578	<p><i>severe (578): No unit number associated with *</i></p> <p>FOR\$IOS_F6312. In an INQUIRE statement, the NUMBER option was specified for the file associated with * (console).</p>
580	<p><i>severe (580): Illegal unit number</i></p> <p>FOR\$IOS_F6314. An illegal unit number was specified.</p> <p>Legal unit numbers can range from 0 through 2**31-1, inclusive.</p>
581	<p><i>severe (581): Illegal RECL value</i></p> <p>FOR\$IOS_F6315. A negative or zero record length was specified for a direct file.</p> <p>The smallest valid record length for direct files is 1.</p>
582	<p><i>severe (582): Array already allocated</i></p> <p>FOR\$IOS_F6316. The program attempted to ALLOCATE an already allocated array.</p>
583	<p><i>severe (583): Array size zero or negative</i></p> <p>FOR\$IOS_F6317. The size specified for an array in an ALLOCATE statement must be greater than zero.</p>
584	<p><i>severe (584): Non-HUGE array exceeds 64K</i></p> <p>FOR\$IOS_F6318.</p>
585	<p><i>severe (585): Array not allocated</i></p> <p>FOR\$IOS_F6319. The program attempted to DEALLOCATE an array that was never allocated.</p>

586	<p><i>severe (586): BACKSPACE illegal on terminal device</i></p> <p>FOR\$IOS_F6400. A BACKSPACE statement specified a unit connected to a terminal device such as a terminal or printer.</p>
587	<p><i>severe (587): EOF illegal on terminal device</i></p> <p>FOR\$IOS_F6401. An EOF intrinsic function specified a unit connected to a terminal device such as a terminal or printer.</p>
588	<p><i>severe (588): ENDFILE illegal on terminal device</i></p> <p>FOR\$IOS_F6402. An ENDFILE statement specified a unit connected to a terminal device such as a terminal or printer.</p>
589	<p><i>severe (589): REWIND illegal on terminal device</i></p> <p>FOR\$IOS_F6403. A REWIND statement specified a unit connected to a terminal device such as a terminal or printer.</p>
590	<p><i>severe (590): DELETE illegal for read-only file</i></p> <p>FOR\$IOS_F6404. A CLOSE statement specified STATUS='DELETE' for a read-only file.</p>
591	<p><i>severe (591): External I/O illegal beyond end of file</i></p> <p>FOR\$IOS_F6405. The program tried to access a file after executing an ENDFILE statement or after it encountered the end-of-file record during a read operation.</p> <p>A BACKSPACE, REWIND, or OPEN statement must be used to reposition the file before execution of any I/O statement that transfers data.</p>
592	<p><i>severe (592): Truncation error: file closed</i></p> <p>FOR\$IOS_F6406.</p>

593	<p><i>severe (593): Terminal buffer overflow</i></p> <p>FOR\$IOS_F6407. More than 131 characters were input to a record of a unit connected to the terminal (keyboard). Note that the operating system may impose additional limits on the number of characters that can be input to the terminal in a single record.</p>
594	<p><i>severe (594): Comma delimiter disabled after left repositioning</i></p> <p>FOR\$IOS_F6408. If you have record lengths that exceed the buffer size associated with the record, (for instance, the record is a file with the buffer set by BLOCKSIZE in the OPEN statement), either you should not do left tabbing within the record, or you should not use commas as field delimiters. This is because commas are disabled as input field delimiters if left tabbing leaves the record positioned in a previous buffer.</p> <p>For example, consider you have a file LONG.DAT that is one continuous record with data fields separated by commas. You then set the buffer associated with the file to 512 bytes, read more than one buffer size of data, tab left to data in the previous buffer, and attempt to read further data, as follows:</p> <pre> INTEGER value(300) OPEN (1, FILE = 'LONG.DAT', BLOCKSIZE = 512) READ (1, 100) (value(i), i = 1, 300) 100 FORMAT (290I2,TL50,10I2) </pre> <p>In this case, error 594 occurs.</p>
599	<p><i>severe (599): File already connected to a different unit</i></p> <p>FOR\$IOS_F6413. The program tried to connect an already connected file to a new unit.</p> <p>A file can be connected to only one unit at a time.</p>
600	<p><i>severe (600): Access not allowed</i></p> <p>FOR\$IOS_F6414.</p> <p>This error can be caused by one of the following:</p> <ul style="list-style-type: none"> • The filename specified in an OPEN statement was a directory.

	<ul style="list-style-type: none"> • An OPEN statement tried to open a read-only file for writing. • The file was opened with SHARE='DENYRW' by another process.
601	<p><i>severe (601): File already exists</i></p> <p>FOR\$IOS_F6415. An OPEN statement specified STATUS='NEW' for a file that already exists.</p>
602	<p><i>severe (602): File not found</i></p> <p>FOR\$IOS_F6416. An OPEN statement specified STATUS='OLD' for a specified file or a directory path that does not exist.</p>
603	<p><i>severe (603): Too many open files</i></p> <p>FOR\$IOS_F6417. The program exceeded the number of open files the operating system allows.</p>
604	<p><i>severe (604): Too many units connected</i></p> <p>FOR\$IOS_F6418. The program exceeded the number of units that can be connected at one time. Units are connected with the OPEN statement.</p>
605	<p><i>severe (605): Illegal structure for unformatted file</i></p> <p>FOR\$IOS_F6419. The file was opened with FORM='UNFORMATTED' and ACCESS='SEQUENTIAL', but its internal physical structure was incorrect or inconsistent. Possible causes: the file was created in another mode or by a non-Fortran program.</p>
606	<p><i>severe (606): Unknown unit number</i></p> <p>FOR\$IOS_F6420. A statement such as BACKSPACE or ENDFILE specified a file that had not yet been opened. (The READ and WRITE statements do not cause this problem because they prompt you for a file if the file has not been opened yet.)</p>
607	<p><i>severe (607): File read-only or locked against writing</i></p> <p>FOR\$IOS_F6421. The program tried to transfer data to a file that was opened in read-only mode or locked against writing.</p> <p>The error message may indicate a CLOSE error when the fault is actually coming from WRITE. This is because the error is not</p>

	discovered until the program tries to write buffered data when it closes the file.
608	<p><i>severe (608): No space left on device</i></p> <p>FOR\$IOS_F6422. The program tried to transfer data to a file residing on a device (such as a hard disk) that was out of storage space.</p>
609	<p><i>severe (609): Too many threads</i></p> <p>FOR\$IOS_F6423. Too many threads were active simultaneously. At most, 32 threads can be active at one time. Close any unnecessary processes or child windows within your application.</p>
610	<p><i>severe (610): Invalid argument</i></p> <p>FOR\$IOS_F6424.</p>
611	<p><i>severe (611): BACKSPACE illegal for SEQUENTIAL write-only files</i></p> <p>FOR\$IOS_F6425. The BACKSPACE statement is not allowed in files opened with MODE='WRITE' (write-only status) because BACKSPACE requires reading the previous record in the file to provide positioning.</p> <p>Resolve the problem by giving the file read access or by avoiding the BACKSPACE statement. Note that the REWIND statement is valid for files opened as write-only.</p>
612	<p><i>severe (612): File not open for reading or file locked</i></p> <p>FOR\$IOS_F6500. The program tried to read from a file that was not opened for reading or was locked.</p>
613	<p><i>severe (613): End of file encountered</i></p> <p>FOR\$IOS_F6501. The program tried to read more data than the file contains.</p>
614	<p><i>severe (614): Positive integer expected in repeat field</i></p> <p>FOR\$IOS_F6502. When the $i*c$ form is used in list-directed input, the i must be a positive integer. For example, consider the following statement:</p>

```
READ(*,*) a, b
```

Input 2*56.7 is accepted, but input 2.1*56.7 returns error 614.

Run-Time Errors 615 Through 656

This section describes in detail Visual Fortran run-time errors 615 through 656.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
615	<p><i>severe (615): Multiple repeat field</i></p> <p>FOR\$IOS_F6503. In list-directed input of the form $i*c$, an extra repeat field was used. For example, consider the following:</p> <pre>READ(*,*) I, J, K</pre> <p>Input of 2*1*3 returns this error. The 2*1 means send two values, each 1; the *3 is an error.</p>
616	<p><i>severe (616): Invalid number in input</i></p> <p>FOR\$IOS_F6504. Some of the values in a list-directed input record were not numeric. For example, consider the following:</p> <pre>READ(*,*) I, J</pre> <p>The preceding statement would cause this error if the input were: 123 'abc'.</p>

617	<p><i>severe (617): Invalid string in input</i></p> <p>FOR\$IOS_F6505. A string item was not enclosed in single quotation marks.</p>
618	<p><i>severe (618): Comma missing in COMPLEX input</i></p> <p>FOR\$IOS_F6506. When using list-directed input, the real and imaginary components of a complex number were not separated by a comma.</p>
619	<p><i>severe (619): T or F expected in LOGICAL read</i></p> <p>FOR\$IOS_F6507. The wrong format was used for the input field for logical data.</p> <p>The input field for logical data consists of optional blanks, followed by an optional decimal point, followed by a T for true or F for false. The T or F may be followed by additional characters in the field, so that .TRUE. and .FALSE. are acceptable input forms.</p>
620	<p><i>severe (620): Too many bytes read from unformatted record</i></p> <p>FOR\$IOS_F6508. The program tried to read more data from an unformatted file than the current record contained. If the program was reading from an unformatted direct file, it tried to read more than the fixed record length as specified by the RECL option. If the program was reading from an unformatted sequential file, it tried to read more data than was written to the record.</p>
621	<p><i>severe (621): H or apostrophe edit descriptor illegal on input</i></p> <p>FOR\$IOS_F6509. Hollerith (H) or apostrophe edit descriptors were encountered in a format used by a READ statement.</p>
622	<p><i>severe (622): Illegal character in hexadecimal input</i></p> <p>FOR\$IOS_F6510. The input field contained a character that was not hexadecimal.</p> <p>Legal hexadecimal characters are 0 - 9 and A - F.</p>

623	<p><i>severe (623): Variable name not found</i></p> <p>FOR\$IOS_F6511. A name encountered on input from a namelist record is not declared in the corresponding NAMELIST statement.</p>
624	<p><i>severe (624): Invalid NAMELIST input format</i></p> <p>FOR\$IOS_F6512. The input record is not in the correct form for namelist input.</p>
625	<p><i>severe (625): Wrong number of array dimensions</i></p> <p>FOR\$IOS_F6513. In namelist input, an array name was qualified with a different number of subscripts than its declaration, or a non-array name was qualified.</p>
626	<p><i>severe (626): Array subscript exceeds allocated area</i></p> <p>FOR\$IOS_F6514. A subscript was specified in namelist input which exceeded the declared dimensions of the array.</p>
627	<p><i>severe (627): Invalid subrange in NAMELIST input</i></p> <p>FOR\$IOS_F6515. A character item in namelist input was qualified with a subrange that did not meet the requirement that $1 \leq e1 \leq e2 \leq \text{len}$ (where "len" is the length of the character item, "e1" is the leftmost position of the substring, and "e2" is the rightmost position of the substring).</p>
628	<p><i>severe (628): Substring range specified on non-CHARACTER item</i></p> <p>FOR\$IOS_F6516. A non-CHARACTER item in namelist input was qualified with a substring range.</p>
629	<p><i>severe (629): Internal file overflow</i></p> <p>FOR\$IOS_F6600. The program either overflowed an internal-file record or tried to write to a record beyond the end of an internal file.</p>
630	<p><i>severe (630): Direct record overflow</i></p> <p>FOR\$IOS_F6601. The program tried to write more than the number of bytes specified in the RECL option to an individual record of a direct-access file.</p>

631	<p><i>severe (631): Numeric field bigger than record size</i></p> <p>FOR\$IOS_F6602. The program tried to write a noncharacter item across a record boundary in list-directed or namelist output. Only character constants can cross record boundaries.</p>
632	<p><i>severe (632): Heap space limit exceeded</i></p> <p>FOR\$IOS_F6700. The program ran out of heap space. The ALLOCATE statement and various internal functions allocate memory from the heap. This error will be generated when the last of the heap space is used up.</p>
633	<p><i>severe (633): Scratch file name limit exceeded</i></p> <p>FOR\$IOS_F6701. The program exhausted the template used to generate unique scratch-file names. The maximum number of scratch files that can be open at one time is 26.</p>
634	<p><i>severe (634): D field exceeds W field in ES edit descriptor</i></p> <p>FOR\$IOS_F6970. The specified decimal length D exceeds the specified total field width W in an ES edit descriptor.</p>
635	<p><i>severe (635): D field exceeds W field in EN edit descriptor</i></p> <p>FOR\$IOS_F6971. The specified decimal length D exceeds the specified total field width W in an EN edit descriptor.</p>
636	<p><i>severe (636): Exponent of 0 not allowed in format</i></p> <p>FOR\$IOS_F6972.</p>
637	<p><i>severe (637): Integer expected in format</i></p> <p>FOR\$IOS_F6980. An edit descriptor lacked a required integer value. For example, consider the following:</p> <pre style="margin-left: 40px;">WRITE(*, 100) I, J 100 FORMAT (I2, TL, I2)</pre> <p>The preceding code will cause this error because an integer is expected after TL.</p>

638	<p><i>severe (638): Initial left parenthesis expected in format</i></p> <p>FOR\$IOS_F6981. A format did not begin with a left parenthesis (().</p>
639	<p><i>severe (639): Positive integer expected in format</i></p> <p>FOR\$IOS_F6982. A zero or negative integer value was used in a format.</p> <p>Negative integer values can appear only with the P edit descriptor. Integer values of 0 can appear only in the <i>d</i> and <i>m</i> fields of numeric edit descriptors.</p>
640	<p><i>severe (640): Repeat count on nonrepeatable descriptor</i></p> <p>FOR\$IOS_F6983. One or more BN, BZ, S, SS, SP, T, TL, TR, /, \$, :, or apostrophe (') edit descriptors had repeat counts associated with them.</p>
641	<p><i>severe (641): Integer expected preceding H, X, or P edit descriptor</i></p> <p>FOR\$IOS_F6984. An integer did not precede a (nonrepeatable) H, X, or P edit descriptor.</p> <p>The correct formats for these descriptors are <i>nH</i>, <i>nX</i>, and <i>kP</i>, respectively, where <i>n</i> is a positive integer and <i>k</i> is an optionally signed integer.</p>
642	<p><i>severe (642): N or Z expected after B in format</i></p> <p>FOR\$IOS_F6985. To control interpretation of embedded and trailing blanks within numeric input fields, you must specify BN (to ignore them) or BZ (to interpret them as zeros).</p>
643	<p><i>severe (643): Format nesting limit exceeded</i></p> <p>FOR\$IOS_F6986. More than 16 sets of parentheses were nested inside the main level of parentheses in a format.</p>
644	<p><i>severe (644): '.' expected in format</i></p> <p>FOR\$IOS_F6987. No period appeared between the <i>w</i> and <i>d</i> fields of a D, E, F, or G edit descriptor.</p>

645	<p><i>severe (645): Unexpected end of format</i></p> <p>FOR\$IOS_F6988. An incomplete format was used.</p> <p>Improperly matched parentheses, an unfinished Hollerith (H) descriptor, or another incomplete descriptor specification can cause this error.</p>
646	<p><i>severe (646): Unexpected character in format</i></p> <p>FOR\$IOS_F6989. A character that cannot be interpreted as part of a valid edit descriptor was used in a format. For example, consider the following:</p> <pre style="margin-left: 40px;">WRITE(*, 100) I, J 100 FORMAT (I2, TL4.5, I2)</pre> <p>The code will generate this error because TL4.5 is not a valid edit descriptor. An integer must follow TL.</p>
647	<p><i>severe (647): M field exceeds W field in I edit descriptor</i></p> <p>FOR\$IOS_F6990. In syntax <i>Iw.m</i>, the value of <i>m</i> cannot exceed the value of <i>w</i>.</p>
648	<p><i>severe (648): Integer out of range in format</i></p> <p>FOR\$IOS_F6991. An integer value specified in an edit descriptor was too large to represent as a 4-byte integer.</p>
649	<p><i>severe (649): format not set by ASSIGN</i></p> <p>FOR\$IOS_F6992. The format specifier in a READ, WRITE, or PRINT statement was an integer variable, but an ASSIGN statement did not properly assign it the statement label of a FORMAT statement in the same program unit.</p>
650	<p><i>severe (650): Separator expected in format</i></p> <p>FOR\$IOS_F6993. Within format specifications, edit descriptors must be separated by commas or slashes (/).</p>
651	<p><i>severe (651): %c or \$: nonstandard edit descriptor in format</i></p> <p>FOR\$IOS_F6994.</p>

652	<p><i>severe (652): Z: nonstandard edit descriptor in format</i></p> <p>FOR\$IOS_F6995. Z is not a standard edit descriptor in format.</p> <p>If you want to transfer hexadecimal values, you must use the edit descriptor form $Zw[m]$, where w is the field width and m is the minimum number of digits that must be in the field (including leading zeros).</p>
653	<p><i>severe (653): DOS graphics not supported under Windows NT</i></p> <p>FOR\$IOS_F6996.</p>
654	<p><i>severe (654): Graphics error</i></p> <p>FOR\$IOS_F6997. An OPEN statement in which IOFOCUS was TRUE, either explicitly or by default, failed because the new window could not receive focus. The window handle may be invalid, or closed, or there may be a memory resource problem.</p>
655	<p><i>severe (655): Using QuickWin is illegal in console application</i></p> <p>FOR\$IOS_F6998. A call to QuickWin from a console application was encountered during execution.</p>
656	<p><i>severe (656): Illegal 'ADVANCE' value</i></p> <p>FOR\$IOS_F6999. The ADVANCE option can only take the values 'YES' and 'NO'. ADVANCE='YES' is the default. ADVANCE is a READ statement option.</p>

Run-Time Errors 657 Through 689

This section describes in detail Visual Fortran run-time errors 657 through 689.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
657	<p><i>severe (657): DIM argument to SIZE out of range</i></p> <p>FOR\$IOS_F6702. The argument specified for DIM must be greater than or equal to 1, and less than or equal to the number of dimensions in the specified array. Consider the following:</p> <pre style="margin-left: 40px;">i = SIZE (array, DIM = dim)</pre> <p>In this case, $1 \leq \text{dim} \leq n$, where n is the number of dimensions in array.</p>
658	<p><i>severe (657): Undefined POINTER used as argument to ASSOCIATED function</i></p> <p>FOR\$IOS_F6703. A POINTER used as an argument to the ASSOCIATED function must be defined; that is, assigned to a target, allocated, or nullified.</p>
659	<p><i>severe (659): Reference to uninitialized POINTER</i></p> <p>FOR\$IOS_F6704. Except in an assignment statement, a pointer must not be referenced until it has been initialized: assigned to a target, allocated or nullified.</p>
660	<p><i>severe (660): Reference to POINTER which is not associated</i></p> <p>FOR\$IOS_F6705. Except in an assignment statement and certain procedure references, a pointer must not be referenced until it has been associated: either assigned to a target or allocated.</p>
661	<p><i>severe (661): Reference to uninitialized POINTER 'pointer'</i></p> <p>FOR\$IOS_F6706. Except in an assignment statement, a pointer must not be referenced until it has been initialized: assigned to a target, allocated or nullified.</p>

662	<p><i>severe (662): reference to POINTER 'pointer' which is not associated</i></p> <p>FOR\$IOS_F6707. Except in an assignment statement and certain procedure references, a pointer must not be referenced until it has been associated: either assigned to a target or allocated.</p>
663	<p><i>severe (663): Out of range: substring starting position 'pos' is less than 1</i></p> <p>FOR\$IOS_F6708. A substring starting position must be a positive integer variable or expression that indicates a position in the string: at least 1 and no greater than the length of the string.</p>
664	<p><i>severe (664): Out of range: substring ending position 'pos' is greater than string length 'len'</i></p> <p>FOR\$IOS_F6709. A substring ending position must be a positive integer variable or expression that indicates a position in the string: at least 1 and no greater than the length of the string.</p>
665	<p><i>severe (665): Subscript 'n' of 'str' (value 'val') is out of range ('first:last')</i></p> <p>FOR\$IOS_F6710. The subscript for a substring within a string is not a valid string position: at least 1 and no greater than the length of the string.</p>
666	<p><i>severe (666): Subscript 'n' of 'str' (value 'val') is out of range ('first:*')</i></p> <p>FOR\$IOS_F6711. The subscript for a substring within a string is not a valid string position: at least 1 and no greater than the length of the string.</p>
667	<p><i>severe (667): VECTOR argument to PACK has incompatible character length</i></p> <p>FOR\$IOS_F6712. The character length of elements in the VECTOR argument to PACK is not the same as the character length of elements in the array to be packed.</p>
668	<p><i>severe (668): VECTOR argument to PACK is too small</i></p> <p>FOR\$IOS_F6713. The VECTOR argument to PACK must have at least as many elements as there are true elements in MASK (the array that controls packing).</p>

669	<p><i>severe (669): SOURCE and PAD arguments to RESHAPE have different character lengths</i></p> <p>FOR\$IOS_F6714. The character length of elements in the SOURCE and PAD arguments to PACK must be the same.</p>
670	<p><i>severe (670): Element 'n' of SHAPE argument to RESHAPE is negative</i></p> <p>FOR\$IOS_F6715. The SHAPE vector specifies the shape of the reshaped array. Since an array cannot have a negative dimension, SHAPE cannot have a negative element.</p>
671	<p><i>severe (671): SOURCE too small for specified SHAPE in RESHAPE, and no PAD</i></p> <p>FOR\$IOS_F6716. If there is no PAD array, the SOURCE argument to RESHAPE must have enough elements to make an array of the shape specified by SHAPE.</p>
672	<p><i>severe (672): Out of memory</i></p> <p>FOR\$IOS_F6717. The system ran out of memory while trying to make the array specified by RESHAPE. If possible, reset your virtual memory size through the Windows Control Panel, or close unnecessary applications and deallocate all allocated arrays that are no longer needed.</p>
673	<p><i>severe (673): SHAPE and ORDER arguments to RESHAPE have different sizes ('size1' and 'size2')</i></p> <p>FOR\$IOS_F6718. ORDER specifies the order of the array dimensions given in SHAPE, and they must be vectors of the same size.</p>
674	<p><i>severe (674): Element 'n' of ORDER argument to RESHAPE is out of range ('range')</i></p> <p>FOR\$IOS_F6719. The ORDER argument specifies the order of the dimensions of the reshaped array, and it must be a permuted list of (1, 2, ..., n) where n is the highest dimension in the reshaped array.</p>

675	<p><i>severe (675): Value 'val' occurs twice in ORDER argument to RESHAPE</i></p> <p>FOR\$IOS_F6720. The ORDER vector specifies the order of the dimensions of the reshaped array, and it must be a permuted list of (1, 2, ..., n) where n is the highest dimension in the reshaped array. No dimension can occur twice.</p>
676	<p><i>severe (676): Impossible nextelt overflow in RESHAPE</i></p> <p>FOR\$IOS_F6721.</p>
677	<p><i>severe (677): Invalid value 'dim' for argument DIM for SPREAD of rank 'rank' source</i></p> <p>FOR\$IOS_F6722. The argument specified for DIM to SPREAD must be greater than or equal to 1, and less than or equal to one larger than the number of dimensions (rank) of SOURCE. Consider the following statement:</p> <pre style="margin-left: 40px;">result = SPREAD (SOURCE= array, DIM = dim, NCOPIES = k)</pre> <p>In this case, $1 \leq \text{dim} \leq n + 1$, where n is the number of dimensions in array.</p>
678	<p><i>severe (678): Complex zero raised to power zero</i></p> <p>FOR\$IOS_F6723. Zero of any type (complex, real, or integer) cannot be raised to zero power.</p>
679	<p><i>severe (679): Complex zero raised to negative power</i></p> <p>FOR\$IOS_F6724. Zero of any type (complex, real, or integer) cannot be raised to a negative power. Raising to a negative power inverts the operand.</p>
680	<p><i>severe (680): Impossible error in NAMELIST input</i></p> <p>FOR\$IOS_F6725.</p>

681	<p><i>severe (681):DIM argument to CSHIFT ('dim') is out of range</i></p> <p>FOR\$IOS_F6726. The optional argument DIM specifies the dimension along which to perform the circular shift, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array to be shifted. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in the array to be shifted.</p>
682	<p><i>severe (682): DIM argument ('dim') to CSHIFT is out of range (1:'n')</i></p> <p>FOR\$IOS_F6727. The optional argument DIM specifies the dimension along which to perform the circular shift, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array to be shifted. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in the array to be shifted.</p>
683	<p><i>severe (683): Shape mismatch (dimension 'dim') between ARRAY and SHIFT in CSHIFT</i></p> <p>FOR\$IOS_F6728. The SHIFT argument to CSHIFT must be either scalar or an array one dimension smaller than the shifted array. If an array, the shape of the SHIFT must conform to the shape of the array being shifted in every dimension except the one being shifted along.</p>
684	<p><i>severe (684): Internal error - bad arguments to CSHIFT_CA</i></p> <p>FOR\$IOS_F6729.</p>
685	<p><i>severe (685): Internal error - bad arguments to CSHIFT_CAA</i></p> <p>FOR\$IOS_F6730.</p>
686	<p><i>severe (686): DATE argument to DATE_AND_TIME is too short (LEN='len')</i></p> <p>FOR\$IOS_F6731. The character DATE argument must have a length of at least 8 to contain the complete value.</p>
687	<p><i>severe (687): TIME argument to DATE_AND_TIME is too short (LEN='len')</i></p> <p>FOR\$IOS_F6732. The character TIME argument must have a length of at least 10 to contain the complete value.</p>

688	<p><i>severe (688): ZONE argument to DATE_AND_TIME is too short (LEN='len')</i></p> <p>FOR\$IOS_F6733. The character ZONE argument must have a length of at least 5 to contain the complete value.</p>
689	<p><i>severe (689): VALUES argument to DATE_AND_TIME is too small ('size' elements)</i></p> <p>FOR\$IOS_F6734. The integer VALUES argument must be a one-dimensional array with a size of at least 8 to hold all returned values.</p>

Run-Time Errors 690 Through 719

This section describes in detail Visual Fortran run-time errors 690 through 719.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
690	<p><i>severe (690): Out of range: DIM argument to COUNT has value 'dim'</i></p> <p>FOR\$IOS_F6735. The optional argument DIM specifies the dimension along which to count true elements of MASK, and must be greater than or equal to 1 and less than or equal to the number of dimensions in MASK. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in MASK.</p>

691	<p><i>severe (691): Out of range: DIM argument to COUNT has value 'dim' with MASK of rank 'rank'</i></p> <p>FOR\$IOS_F6736. The optional argument DIM specifies the dimension along which to count true elements of MASK, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in MASK. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in MASK.</p>
692	<p><i>severe (692): Out of range: DIM argument to PRODUCT has value 'dim'</i></p> <p>FOR\$IOS_F6737. The optional argument DIM specifies the dimension along which to compute the product of elements in an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be multiplied.</p>
693	<p><i>severe (693): Out of range: DIM argument to PRODUCT has value 'dim' with ARRAY of rank 'rank'</i></p> <p>FOR\$IOS_F6738. The optional argument DIM specifies the dimension along which to compute the product of elements in an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) of the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be multiplied.</p>
694	<p><i>severe (694): Out of range: DIM argument to SUM has value 'dim' with ARRAY of rank 'rank'</i></p> <p>FOR\$IOS_F6739. The optional argument DIM specifies the dimension along which to sum the elements of an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) of the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be summed.</p>
695	<p><i>severe (695): Real zero raised to zero power</i></p> <p>FOR\$IOS_F6740. Zero of any type (real, complex, or integer) cannot be raised to zero power.</p>

696	<p><i>severe (696): Real zero raised to negative power</i></p> <p>FOR\$IOS_F6741. Zero of any type (real, complex, or integer) cannot be raised to a negative power. Raising to a negative power inverts the operand.</p>
697	<p><i>severe (697): Out of range: DIM argument to SUM has value 'dim'</i></p> <p>FOR\$IOS_F6742. The optional argument DIM specifies the dimension along which to sum the elements of an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be summed.</p>
698	<p><i>severe (698): DIM argument ('dim') to EOSHIFT is out of range (1:'n')</i></p> <p>FOR\$IOS_F6743. The optional argument DIM specifies the dimension along which to perform an end-off shift in an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be shifted.</p>
699	<p><i>severe (699): Shape mismatch (dimension 'dim') between ARRAY and BOUNDARY in EOSHIFT</i></p> <p>FOR\$IOS_F6744. The BOUNDARY argument to EOSHIFT must be either scalar or an array one dimension smaller than the shifted array. If an array, the shape of the BOUNDARY must conform to the shape of the array being shifted in every dimension except the one being shifted along.</p>
700	<p><i>severe (700): DIM argument to EOSHIFT is out of range ('dim')</i></p> <p>FOR\$IOS_F6745. The optional argument DIM specifies the dimension along which to perform an end-off shift in an array, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array holding the elements to be shifted.</p>

701	<p><i>severe (701): Shape mismatch (dimension 'dim') between ARRAY and SHIFT in EOSHIFT</i></p> <p>FOR\$IOS_F6746. The SHIFT argument to EOSHIFT must be either scalar or an array one dimension smaller than the shifted array. If an array, the shape of the SHIFT must conform to the shape of the array being shifted in every dimension except the one being shifted along.</p>
702	<p><i>severe (702): BOUNDARY argument to EOSHIFT has wrong LEN ('len1 instead of len2')</i></p> <p>FOR\$IOS_F6747. The character length of elements in the BOUNDARY argument and in the array being end-off shifted must be the same.</p>
703	<p><i>severe (703): BOUNDARY has LEN 'len' instead of 'len' to EOSHIFT</i></p> <p>FOR\$IOS_F6748.</p>
704	<p><i>severe (704): Internal error - bad arguments to EOSHIFT</i></p> <p>FOR\$IOS_F6749.</p>
705	<p><i>severe (705): GETARG: value of argument 'num' is out of range</i></p> <p>FOR\$IOS_F6750. The value used for the number of the command-line argument to retrieve with GETARG must be 0 or a positive integer. If the number of the argument to be retrieved is greater than the actual number of arguments, blanks are returned, but no error occurs.</p>
706	<p><i>severe (706): FLUSH: value of LUNIT 'num' is out of range</i></p> <p>FOR\$IOS_F6751. The unit number specifying which I/O unit to flush to its associated file must be an integer between 0 and 2**31-1, inclusive. If the unit number is valid, but the unit is not opened, error F6752 is generated.</p>
707	<p><i>severe (707): FLUSH: Unit 'n' is not connected</i></p> <p>FOR\$IOS_F6752. The I/O unit specified to be flushed to its associated file is not connected to a file.</p>

708	<p><i>severe (708): Invalid string length ('len') to ICHAR</i></p> <p>FOR\$IOS_F6753. The character argument to ICHAR must have length 1.</p>
709	<p><i>severe (709): Invalid string length ('len') to IACHAR</i></p> <p>FOR\$IOS_F6754. The character argument to IACHAR must have length 1.</p>
710	<p><i>severe (710): Integer zero raised to negative power</i></p> <p>FOR\$IOS_F6755. Zero of any type (integer, real, or complex) cannot be raised to a negative power. Raising to a negative power inverts the operand.</p>
711	<p><i>severe (711): INTEGER zero raised to zero power</i></p> <p>FOR\$IOS_F6756. Zero of any type (integer, real, or complex) cannot be raised to zero power.</p>
712	<p><i>severe (712): SIZE argument ('size') to ISHFTC intrinsic out of range</i></p> <p>FOR\$IOS_F6757. The argument SIZE must be positive and must not exceed the bit size of the integer being shifted. The bit size of this integer can be determined with the function BIT_SIZE.</p>
713	<p><i>severe (713): SHIFT argument ('shift') to ISHFTC intrinsic out of range</i></p> <p>FOR\$IOS_F6758. The argument SHIFT to ISHFTC must be an integer whose absolute value is less than or equal to the number of bits being shifted: either all bits in the number being shifted or a subset specified by the optional argument SIZE.</p>
714	<p><i>severe (714): Out of range: DIM argument to LBOUND has value 'dim'</i></p> <p>FOR\$IOS_F6759. The optional argument DIM specifies the dimension whose lower bound is to be returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>

715	<p><i>severe (715): Out of range: DIM argument ('dim') to LBOUND greater than ARRAY rank 'rank'</i></p> <p>FOR\$IOS_F6760. The optional argument DIM specifies the dimension whose lower bound is to be returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
716	<p><i>severe (716): Out of range: DIM argument to MAXVAL has value 'dim'</i></p> <p>FOR\$IOS_F6761. The optional argument DIM specifies the dimension along which maximum values are returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
717	<p><i>severe (717): Out of range: DIM argument to MAXVAL has value 'dim' with ARRAY of rank 'rank'</i></p> <p>FOR\$IOS_F6762. The optional argument DIM specifies the dimension along which maximum values are returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
718	<p><i>severe (718): Cannot allocate temporary array -- out of memory</i></p> <p>FOR\$IOS_F6763. There is not enough memory space to hold a temporary array.</p> <p>Dynamic memory allocation is limited by several factors, including swap file size and memory requirements of other applications that are running. If you encounter an unexpectedly low limit, you might need to reset your virtual memory size through the Windows Control Panel or redefine the swap file size. Allocated arrays that are no longer needed should be deallocated.</p>
719	<p><i>severe (719): Attempt to DEALLOCATE part of a larger object</i></p> <p>FOR\$IOS_F6764. An attempt was made to DEALLOCATE a pointer to an array subsection or an element within a derived type. The whole data object must be deallocated; parts cannot be deallocated.</p>

Run-Time Errors 720 Through 759

This section describes in detail Visual Fortran run-time errors 720 through 759.

In the following table, the first column lists error numbers returned to IOSTAT variables when an I/O error is detected.

The first line of the second column provides the message as it is displayed (following *forrtl:*), including the severity level, message number, and the message text. The following lines of the second column contain the status condition symbol (such as FOR\$IOS_INCRECTYP) and an explanation of the message.

For information about severity levels, see the [Introduction](#).

Number	Severity Level, Number, and Message Text; Condition Symbol and Explanation
720	<p><i>severe (720): Pointer in DEALLOCATE is ASSOCIATED with an ALLOCATABLE array</i></p> <p>FOR\$IOS_F6765. Deallocating a pointer associated with an allocatable target is illegal. Instead, deallocate the target the pointer points to, which frees memory and disassociates the pointer.</p>
721	<p><i>severe (721): Attempt to DEALLOCATE an object which was not allocated</i></p> <p>FOR\$IOS_F6766. You cannot deallocate an array unless it has been previously allocated. You cannot deallocate a pointer whose target was not created by allocation. The intrinsic function ALLOCATED can be used to determine whether an allocatable array is currently allocated.</p>
722	<p><i>severe (722): Cannot ALLOCATE scalar POINTER -- out of memory</i></p> <p>FOR\$IOS_F6767. There is not enough memory space to allocate the pointer.</p> <p>Dynamic memory allocation is limited by several factors, including swap file size and memory requirements of other applications that are running. If you encounter an unexpectedly low limit, you might need to reset your virtual memory size through the Windows Control Panel or redefine the swap file size. Allocated arrays that are no longer needed should be</p>

	deallocated.
723	<p><i>severe (723): DEALLOCATE: object not allocated/associated</i></p> <p>FOR\$IOS_F6768. You cannot deallocate an array unless it has been previously allocated. You cannot deallocate a pointer whose target was not created by allocation, or a pointer that has undefined association status.</p> <p>The intrinsic function ALLOCATED can be used to determine whether an allocatable array is currently allocated.</p>
724	<p><i>severe (724): Cannot ALLOCATE POINTER array -- out of memory</i></p> <p>FOR\$IOS_F6769. There is not enough memory space to allocate the POINTER array.</p> <p>Dynamic memory allocation is limited by several factors, including swap file size and memory requirements of other applications that are running. If you encounter an unexpectedly low limit, you might need to reset your virtual memory size through the Windows Control Panel or redefine the swap file size. Allocated arrays that are no longer needed should be deallocated.</p>
725	<p><i>severe (725): DEALLOCATE: Array not allocated</i></p> <p>FOR\$IOS_F6770. It is illegal to DEALLOCATE an array that is not allocated. You can check the allocation status of an array before deallocating with the ALLOCATED function.</p>
726	<p><i>severe (726): DEALLOCATE: Character array not allocated</i></p> <p>FOR\$IOS_F6771. It is illegal to DEALLOCATE an array that is not allocated. You can check the allocation status of an array before deallocating with the ALLOCATED function.</p>
727	<p><i>severe (727): Cannot ALLOCATE allocatable array -- out of memory</i></p> <p>FOR\$IOS_F6772. There is not enough memory space to hold the array.</p> <p>Dynamic memory allocation is limited by several factors, including swap file size and memory requirements of other applications that are running. If you encounter an unexpectedly low limit, you might need to reset your virtual memory size through the Windows Control Panel or redefine the</p>

	<p>swap file size. Allocated arrays that are no longer needed should be deallocated.</p>
728	<p><i>severe (728): Cannot allocate automatic object -- out of memory</i></p> <p>FOR\$IOS_F6773. There is not enough memory space to hold the automatic data object.</p> <p>Dynamic memory allocation is limited by several factors, including swap file size and memory requirements of other applications that are running. If you encounter an unexpectedly low limit, you might need to reset your virtual memory size through the Windows Control Panel or redefine the swap file size. Allocated arrays that are no longer needed should be deallocated.</p> <p>An automatic data object is an object that is declared in a procedure subprogram or interface, is not a dummy argument, and depends on a nonconstant expression. For example:</p> <pre> SUBROUTINE EXAMPLE (N) DIMENSION A (N, 5), B(10*N) </pre> <p>The arrays A and B in the example are automatic data objects.</p>
729	<p><i>severe (729): DEALLOCATE failure: ALLOCATABLE array is not ALLOCATED</i></p> <p>FOR\$IOS_F6774. It is illegal to DEALLOCATE an array that is not allocated. You can check the allocation status of an array before deallocating with the ALLOCATED function.</p>
730	<p><i>severe (730): Out of range: DIM argument to MINVAL has value 'dim'</i></p> <p>FOR\$IOS_F6775. The optional argument DIM specifies the dimension along which minimum values are returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>

731	<p><i>severe (731): Out of range: DIM argument to MINVAL has value 'dim' with ARRAY of rank 'rank'</i></p> <p>FOR\$IOS_F6776. The optional argument DIM specifies the dimension along which minimum values are returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
732	<p><i>severe (732): P argument to MOD is double precision zero</i></p> <p>FOR\$IOS_F6777. MOD(A,P) is computed as $A - \text{INT}(A/P) * P$. So, P cannot be zero.</p>
733	<p><i>severe (733): P argument to MOD is integer zero</i></p> <p>FOR\$IOS_F6778. MOD(A,P) is computed as $A - \text{INT}(A/P) * P$. So, P cannot be zero.</p>
734	<p><i>severe (734): P argument to MOD is real zero</i></p> <p>FOR\$IOS_F6779. MOD(A,P) is computed as $A - \text{INT}(A/P) * P$. So, P cannot be zero.</p>
735	<p><i>severe (735): P argument to MODULO is real zero</i></p> <p>FOR\$IOS_F6780. MODULO(A,P) for real numbers is computed as $A - \text{FLOOR}(A/P) * P$. So, P cannot be zero.</p>
736	<p><i>severe (736): P argument to MODULO is zero</i></p> <p>FOR\$IOS_F6781. In the function, MODULO(A,P), P cannot be zero.</p>
737	<p><i>severe (737): Argument S to NEAREST is zero</i></p> <p>FOR\$IOS_F6782. The sign of the S argument to NEAREST(X,S) determines the direction of the search for the nearest number to X, and cannot be zero.</p>
738	<p><i>severe (738): Heap storage exhausted</i></p> <p>FOR\$IOS_F6783.</p>

739	<p><i>severe (739): PUT argument to RANDOM_SEED is too small</i></p> <p>FOR\$IOS_F6784. The integer array PUT must be greater than or equal to the number of integers the processor uses to set the seed value. This number can be determined by calling RANDOM_SEED with the SIZE argument. For example:</p> <pre> INTEGER, ALLOCATABLE SEED CALL RANDOM_SEED() ! initialize processor CALL RANDOM_SEED(SIZE = K) ! get size of seed ALLOCATE SEED(K) ! allocate array CALL RANDOM_SEED(PUT = SEED) ! set the seed </pre> <p>Note that RANDOM_SEED can be called with at most one argument at a time.</p>
740	<p><i>severe (740): GET argument to RANDOM_SEED is too small</i></p> <p>FOR\$IOS_F6785. The integer array GET must be greater than or equal to the number of integers the processor uses to set the seed value. This number can be determined by calling RANDOM_SEED with the SIZE argument. For example:</p> <pre> INTEGER, ALLOCATABLE SEED CALL RANDOM_SEED() ! initialize processor CALL RANDOM_SEED(SIZE = K) ! get size of seed ALLOCATE SEED(K) ! allocate array CALL RANDOM_SEED(GET = SEED) ! get the seed </pre> <p>Note that RANDOM_SEED can be called with at most one argument at a time.</p>
741	<p><i>severe (741): Recursive i/o reference</i></p> <p>FOR\$IOS_F6786.</p>
742	<p><i>severe (742): Argument to SHAPE intrinsic is not PRESENT</i></p> <p>FOR\$IOS_F6787.</p>

743	<p><i>severe (743): Out of range: DIM argument to UBOUND had value 'dim'</i></p> <p>FOR\$IOS_F6788. The optional argument DIM specifies the dimension whose upper bound is to be returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
744	<p><i>severe (744): DIM argument ('dim') to UBOUND greater than ARRAY rank 'rank'</i></p> <p>FOR\$IOS_F6789. The optional argument DIM specifies the dimension whose upper bound is to be returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
745	<p><i>severe (745): Out of range: UBOUND of assumed-size array with DIM==rank ('rank')</i></p> <p>FOR\$IOS_F6790. The optional argument DIM specifies the dimension whose upper bound is to be returned.</p> <p>An assumed-size array is a dummy argument in a subroutine or function, and the upper bound of its last dimension is determined by the size of actual array passed to it. Assumed-size arrays have no determined shape, and you cannot use UBOUND to determine the extent of the last dimension. You can use UBOUND to determine the upper bound of one of the fixed dimensions, in which case you must pass the dimension number along with the array name.</p>
746	<p><i>severe (746): Out of range: DIM argument ('dim') to UBOUND greater than ARRAY rank</i></p> <p>FOR\$IOS_F6791. The optional argument DIM specifies the dimension whose upper bound is to be returned, and must be greater than or equal to 1 and less than or equal to the number of dimensions (rank) in the array. That is, $1 \leq \text{DIM} \leq n$, where n is the number of dimensions in array.</p>
747	<p><i>severe (747): Shape mismatch: Dimension 'shape' extents are 'ext1' and 'ext2'</i></p> <p>FOR\$IOS_F6792.</p>

748	<p><i>severe (748): Illegal POSITION value</i></p> <p>FOR\$IOS_F6793. An illegal value was used with the POSITION specifier.</p> <p>POSITION accepts the following values:</p> <ul style="list-style-type: none">• 'ASIS' (the default)• 'REWIND' - on Fortran I/O systems, this is the same as 'ASIS'• 'APPEND'
749	<p><i>severe (749): Illegal ACTION value</i></p> <p>FOR\$IOS_F6794. An illegal value was used with the ACTION specifier.</p> <p>ACTION accepts the following values:</p> <ul style="list-style-type: none">• 'READ'• 'WRITE'• 'READWRITE' - the default
750	<p><i>severe (750): DELIM= specifier not allowed for an UNFORMATTED file</i></p> <p>FOR\$IOS_F6795. The DELIM specifier is only allowed for files connected for formatted data transfer. It is used to delimit character constants in list-directed and namelist output.</p>
751	<p><i>severe (751): Illegal DELIM value</i></p> <p>FOR\$IOS_F6796. An illegal value was used with the DELIM specifier.</p> <p>DELIM accepts the following values:</p> <ul style="list-style-type: none">• 'APOSTROPHE'• 'QUOTE'• 'NONE' - the default

752	<p><i>severe (752): PAD= specifier not allowed for an UNFORMATTED file</i></p> <p>FOR\$IOS_F6797. The PAD specifier is only allowed for formatted input records. It indicates whether the formatted input record is padded with blanks when an input list and format specification requires more data than the record contains.</p>
753	<p><i>severe (753): Illegal PAD= value</i></p> <p>FOR\$IOS_F6798. An illegal value was used with the PAD specifier.</p> <p>PAD accepts the following values:</p> <ul style="list-style-type: none"> • 'NO' • 'YES' - the default
754	<p><i>severe (754): Illegal CARRIAGECONTROL= value</i></p> <p>FOR\$IOS_F6799. An illegal value was used with the CARRIAGECONTROL specifier.</p> <p>CARRIAGECONTROL accepts the following values:</p> <ul style="list-style-type: none"> • 'FORTRAN' - default if /vms is specified or the unit is connected to a terminal or console • 'LIST' - default for formatted files • 'NONE' - default for unformatted files
755	<p><i>severe (755): SIZE= specifier only allowed with ADVANCE='NO'</i></p> <p>FOR\$IOS_F6800. The SIZE specifier can only appear in a formatted, sequential READ statement that has the specifier ADVANCE='NO' (indicating nonadvancing input).</p>
756	<p><i>severe (756): Illegal character in binary input</i></p> <p>FOR\$IOS_F6801.</p>
757	<p><i>severe (757): Illegal character in octal input</i></p> <p>FOR\$IOS_F6802.</p>

758	<i>severe (758): End of record encountered</i> FOR\$IOS_F6803.
759	<i>severe (759): Illegal subscript in namelist input record</i> FOR\$IOS_F6804.

Linker Tools Messages

The following topics cover linker tools errors and warnings (LNKnnnn), which can be generated LINK, LIB, DUMPBIN, and EDITBIN:

- [Linker Errors LNK1000 to LNK1149](#)
 - [Linker Errors LNK1152 to LNK1189](#)
 - [Linker Errors LNK1190 to LNK1581](#)
 - [Linker Errors LNK2001 to LNK2014](#)
 - [Linker Warnings LNK4001 to LNK4039](#)
 - [Linker Warnings LNK4040 to LNK4069](#)
 - [Linker Warnings LNK4070 to LNK4099](#)
 - [Linker Warnings LNK4102 to LNK4549](#)
 - [Linker Warnings LNK6001 to LNK6026](#)
 - [Corrupt Object File Explanation](#)
-

Corrupt Object File Explanation:

Sometimes the linker is unable to process your object files because they are corrupt. Besides the usual causes of corrupt files such as network problems or exposing a floppy disk to a magnetic field, sometimes the compiler writes an incorrect object file, which you should report to technical support.

After you have ruled out a hardware, operating system, or network problem as the cause of the corrupt object file, you should try the tips below because sometimes by changing compiler options and recompiling, the compiler will write a non-corrupt object file. Here are some things to try when recompiling:

- Turn off optimization with the `/nooptimize (/Od, Disable)` option.
- Use a different code generation option. See the `/tune (/G, Optimize for Processor)` options.
- Change the order of functions and global variables.

With Visual C++ mixed-language programs:

- Disable incremental compilation with the `/Gi-` (Enable Incremental Compilation) option.
- Disable minimal rebuild with the `/Gm-` (Enable Minimal Rebuild) option.
- Compile with the `/Gy` (Enable Function-Level Linking) option to package functions.

Linker Errors LNK1000 to LNK1149

Linker Tools Error LNK1000

[unknown error; consult documentation for technical support options](#)

Note the circumstances of the error, try to isolate the problem and create a reproducible test case, then contact technical support.

Linker Tools Error LNK1101

[incorrect MSPDBxx.DLL version; recheck installation of this product](#)

The version of MSPDBxx.DLL available on your system does not match the version required by this tool.

Linker Tools Error LNK1102

[out of memory](#)

There was not enough memory for the tool to run. Probably the paging file exceeded available disk space. If a shortage of disk space is not the cause, note the circumstances of the error, try to isolate the problem and create a reproducible test case, then request technical support.

Linker Tools Error LNK1103

[debugging information corrupt; recompile module](#)

Probably the compilation was terminated before a valid object file was created. Recompile the given object file. If recompiling does not correct the problem, note the circumstances of the error, try to isolate the problem and create a reproducible test case, then consult technical support. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1104

[cannot open file "*filename*"](#)

The tool could not open the given file. One of the following may be a cause:

- There was not enough disk space.
- The file does not exist.
- The filename or its path was incorrectly specified.
- The specified drive is invalid.
- The file does not have the appropriate permissions.
- The path for filename expands to more than 260 characters.
- If the given file is named LNK*n*, which is a filename generated by the linker for a temporary file, then the directory specified in the TMP environment variable may not exist, or more than one directory is specified for the TMP environment variable. (Only one directory path should be specified for the TMP environment variable.)
- If the error occurs on the executable filename, an earlier version of the executable may still be running. You will need to terminate the executable before linking it. In Windows NT (including Windows 2000) or Windows 95, you can use the utility PVIEW to look for and kill instances of the application.
- If the error message occurs for a library name, and you recently ported the .MAK file from a previous Microsoft Visual C++ development

environment, the library may no longer be valid. Ensure that the library still exists in this circumstance.

- If the error message occurs for a library name such as `CONSOLE.LIB`, and you included object modules from an older Fortran development environment such as Microsoft Fortran PowerStation, recompile all source files with Visual Fortran to create new object files and libraries.

Linker Tools Error LNK1105

cannot close file "*filename*"

The tool could not close the given file. Possibly there was insufficient disk space available.

Linker Tools Error LNK1106

invalid file or disk full: cannot seek to location

The tool could not read or write to location in a memory-mapped file:

- Your disk may be too full to complete the link. Free up some space and try the link again.
- The error may be a result of trying to link over a network. Some networks do not fully support the memory-mapped files used by the linker. Try linking on your local disk to see if that fixes the problem.
- There may be a bad block on your disk. Although the operating system and disk hardware should have detected such an error, you may want to run a disk checking program.

Linker Tools Error LNK1107

invalid or corrupt file: cannot read at location

The tool could not read the file. Recreate the file.

Linker Tools Error LNK1108

cannot write file at location

The tool could not write to the file. One of the following may be a cause:

- There was not enough disk space to create the file.
- The drive being written to was not available, possibly due to a network problem.

Linker Tools Error LNK1109

cannot remove file "*filename*"

LIB could not delete the given file. Before LIB writes the new version of a library, it removes the existing library file. One of the following may be a cause:

- The given file does not have the appropriate permissions.
- The drive containing the file was not available, possibly due to a network problem.

Linker Tools Error LNK1110

cannot rename file "*filename*"

LIB could not rename the given file. When LIB builds a new version of a library, it creates a temporary file, then renames the file. One of the following may be a cause:

- The given file does not have the appropriate permissions.
- The drive containing the file was not available, possibly due to a network problem.

Linker Tools Error LNK1111

invalid /BASE argument "*argument*"

The /BASE option was incorrectly specified. Either no argument was specified or argument is not a number.

Linker Tools Error LNK1112

module machine type "*type1*" conflicts with target machine type "*type2*"

The object files specified as input were compiled for different machine types.

Linker Tools Error LNK1113

invalid machine type

The machine type specified in the object header is not valid. Probably the file is corrupt. Rebuild the object. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1115

/MACHINE option required

LINK or LIB cannot determine the machine type for objects that are provided entirely from libraries. Either specify the /MACHINE option with the appropriate machine type or specify at least one object file in the input.

Linker Tools Error LNK1117

syntax error in option "*option*"

The given option was not correctly specified.

Linker Tools Error LNK1118

syntax error in keyword statement

The given module-definition statement was not correctly specified.

Linker Tools Error LNK1119

invalid ordinal number "*argument*"

The argument following the at sign (@) in an ordinal specification was not a valid number. An ordinal number is an optional argument in either an /EXPORT option in a LINK or LIB command or an EXPORTS statement in a module-definition file. It is an index into the exports table. The number must be an integer in the range 1-65535.

Linker Tools Error LNK1120

number unresolved externals

Error LNK1120 gives you a count (number) of unresolved externals for this link. The conditions that cause unresolved externals are described with error LNK2001 (see [Linker Errors LNK2001 to LNK2014](#)), which precedes this error message (once for each unresolved external).

Linker Tools Error LNK1121

duplicate ordinal number "*number*"

The given ordinal number was specified more than once in either an /EXPORT option in a LINK or LIB command or an EXPORTS statement in a module-definition file. Ordinal numbers must be unique integers in the range 1-65535.

Linker Tools Error LNK1123

failure during conversion to COFF: file invalid or corrupt

The object or resource could not be converted to COFF. This tool requires the format of all input files to be COFF. If an input file is not COFF, the tool runs a conversion tool, either CVTOMF (to convert 32-bit OMF objects) or CVTRES (to convert resource files). This error is preceded by errors or warnings from the conversion tool. One of the following may be a cause:

- The file is corrupt.
- The file is not a valid file type. An example of an invalid type is a 16-bit OMF object.

Linker Tools Error LNK1127

library is corrupt

The library file is corrupt. Rebuild the library.

Linker Tools Error LNK1129

cannot find resolution for weak extern symbol

The given weak external symbol does not have a default resolution. Probably the symbol table is corrupt. Rebuild the object file. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1130

Linker miscalc (base relocations off by number)

There was not enough space in the image to write base relocations.

Linker Tools Error LNK1131

no library file specified

The LIB /EXTRACT command required a library as input, but a library filename was not specified.

Linker Tools Error LNK1132

invalid format for MS-DOS stub file "*filename*"

The filename specified with the /STUB option was not a valid real-mode MS-DOS executable (.EXE) file.

Linker Tools Error LNK1136

invalid or corrupt file

The input file either has a corrupt header or is zero size or abnormally small. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1137

invalid argument specified with /SECTION

Either the name or the attributes argument to the /SECTION option is specified incorrectly.

Linker Tools Error LNK1140

too many modules for program database; relink with /PDB:NONE

The project contains more than 4096 modules. One of the following is a solution:

- Relink using /PDB:NONE.
- Compile some modules without debugging information.
- Reduce the number of modules.

Linker Tools Error LNK1141

failure during build of exports file

LINK could not build the exports (.EXP) file. Causes of this error include system problems such as insufficient memory and syntax errors in options or module-definition statements. This error is preceded by another error that gives additional information.

Linker Tools Error LNK1143

invalid or corrupt file: no symbol for COMDAT section number

The object file is corrupt. Rebuild the file. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1144

error with LINK_REPRO var; cannot open "*filename*"

The linker could not create filename. One of the following may be a cause:

- Check that you've set your LINK_REPRO environment variable to an existing directory and not the current directory.
- Make sure there are no read-only files in the directory.
- Make sure none of the files in the LINK_REPRO directory are open by another process, because the linker needs to write to them.

Linker Tools Error LNK1145

/MERGE created circular link for section "*section*"

You attempted to merge a section into itself. Check the /MERGE options.

Linker Tools Error LNK1146

no argument specified with option "*option*"

The given option requires an argument.

Linker Tools Error LNK1147

invalid number specified with option "*option*"

The argument to the given option was specified incorrectly.

Linker Tools Error LNK1148

failure during conversion to COFF: cannot copy temp file filename

EDITBIN could not convert the input file to COFF. When EDITBIN converts a file, it creates a temporary file, then copies the file. One of the following may be a cause:

- The given file does not have the appropriate permissions.
- There was not enough disk space to create the file.
- The drive being written to was not available, possibly due to a network problem.

Linker Tools Error LNK1149

output filename matches input filename "*filename*"

The output filename specified with the /OUT or /IMPLIB option was the same as an input file.

[Return to Main Linker Error Page](#)

Linker Errors LNK1152 to LNK1189

Linker Tools Error LNK1152

cannot resolve one or more undecorated symbols

This error is preceded by one warning LNK4022 for each undecorated symbol that could not be resolved and by at least two warnings LNK4006 for the duplicate symbols found for the undecorated symbol.

Linker Tools Error LNK1153

/VXD command-line option required

You attempted to build a virtual device driver without the /VXD option. Relink with the /VXD option.

Linker Tools Error LNK1154

specified import library filename matches exports file "*filename*"

The *filename* specified with LINK /IMPORT or with LIB /DEF /OUT conflicted with the filename given by LINK or LIB to the exports file, which is formed from the base name of the main output file and the extension .EXP.

Linker Tools Error LNK1155

special symbol "*symbol*" already defined

The given symbol is reserved for use by LINK.

Linker Tools Error LNK1156

.sbss section not supported

An object file contained an *.sbss* section.

Linker Tools Error LNK1157

fixup overflow; offset of target symbol "symbol" greater than +-8MB

The VXD is too large, or the sections are not arranged properly.

Linker Tools Error LNK1158

cannot run "*filename*"

The given executable file called by LINK is not in the directory that contains LINK and is not in a directory specified in the PATH environment variable.

Linker Tools Error LNK1159

no output file specified

No name was specified for the main output file (executable file or DLL). LINK derives the default name of the output file from the base name of the first object file. If no object files are specified, and if the /OUT option is not used, this error occurs.

Linker Tools Error LNK1160

library with zero objects not allowed

An attempt was made to remove an object from a library that contained only that object. The object was not removed.

Linker Tools Error LNK1161

invalid export specification

Either the /EXPORT option or the EXPORTS module-definition statement incorrectly specified an export. A possible cause is a typing error.

Linker Tools Error LNK1162

expected aux symbol for comdat section number

The linker expected to find an auxiliary symbol table for the indicated COMDAT but could not. The object file is probably corrupt. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1163

invalid selection for comdat section number

The byte in the object file indicating the type of COMDAT section is invalid. The object file is probably corrupt. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1164

section *section* alignment (number) greater than /ALIGN value

The alignment size for the given section in the object file exceeds the value specified with the /ALIGN option. The /ALIGN value must be a power of 2 and

must equal or exceed the section alignment given in the object file. Either recompile with a smaller section alignment or increase the /ALIGN value.

Linker Tools Error LNK1165

link failed because of fixup errors

The build failed due to fixup errors. The /FORCE or /FORCE:UNRESOLVED option overrides this error.

Linker Tools Error LNK1166

cannot adjust code at offset=*offset*, va=*value*

LINK was unable to pad the code as required. Certain instructions are not allowed to cross page boundaries on some processors. LINK attempts to add pads to correct this situation. In this case, LINK could not work around the problem.

Linker Tools Error LNK1167

file contains relocs but header has no machine type

A converted COFF object did not have a machine type specified in its header. One cause of this error is omitting a machine type when converting a .RES file in a separate step before linking.

Linker Tools Error LNK1168

cannot open filename for writing

The given file does not have write permission.

Linker Tools Error LNK1169

one or more multiply defined symbols found

The build failed due to multiple definitions of one or more symbols. This error is preceded by error LNK2005. The /FORCE or /FORCE:MULTIPLE option overrides this error.

Linker Tools Error LNK1170

line in command file contains limit or more characters

The length of a line in a command file must be less than the given limit.

Linker Tools Error LNK1171

unable to load filename

The given DLL was unavailable. The possible locations for the DLL are the current directory, the system directory, the Windows directory, and the directories specified in the PATH environment variable.

Linker Tools Error LNK1172

more than one object with the name "*object*" found; rename object(s) or relink /PDB:NONE

A library contained two or more objects with the same name. Do one of the following:

- Rename the objects using unique names and rebuild the library.
- Link using the /PDB:NONE option.

Linker Tools Error LNK1173

unable to find entrypoint function in filename

The given function does not exist in the given DLL.

Linker Tools Error LNK1174

unable to /REBASE filename; not a valid Win32 image

The format of the given file was invalid.

Linker Tools Error LNK1175

failed to /REBASE filename

The rebase operation failed on the given file.

Linker Tools Error LNK1177

TOC size limit exceeded

The linker was unable to create a TOC (Table of Contents) in your image file. The limit is 2048 entries in the TOC.

Linker Tools Error LNK1178

missing MODEND record; file is invalid or corrupt

The linker tool tried to convert an OMF object module to COFF format but could not find an expected MODEND (module end) record in the OMF object. The OMF object module is corrupt and needs to be recreated or recopied. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1179

invalid or corrupt file: duplicate comdat *comdat*

An object module contained two or more COMDATs with the same name. One possible cause is if you use the Visual C++ /H option with the Visual C++ /Gy option. The /H option limits the length of external names, and the /Gy option packages functions in COMDATs.

Linker Tools Error LNK1180

insufficient disk space to complete link

The linker tool was unable to complete a file operation since the operating system reported that the disk is full. Free up space on your local and network drives (if you are writing files there).

Linker Tools Error LNK1181

cannot open input file "*filename*"

The linker tool could not find "*filename*" because it does not exist or the path was not found.

Linker Tools Error LNK1182

cannot have more than 64K exports

You have reached the linker's limit of 65,536 exports.

Linker Tools Error LNK1183

invalid or corrupt file: extended relocation count number less than 65535

The COFF object file is corrupt since it contains an extended relocation count that is less than 0xFFFF. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1184

invalid section name "*section*" specified in option or directive "*option*"

You gave the linker an invalid section name in option. Invalid section names contain "\$" or blanks.

Linker Tools Error LNK1185

invalid section name "*section*" specified

You gave the linker an invalid section name in a .DEF file. Invalid section names contain "\$" or blanks.

Linker Tools Error LNK1186

invalid or corrupt COFF object; reloc to undefined static symbol "*symbol*"

Your COFF object module is corrupt since it contains a relocation entry for an undefined static symbol. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1187

Corrupt object - unmatched name relocation; ignored

Your object file contains bad relocation information. Recopy or recreate the object file. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK1188

BADFIXUPSECTION:: invalid fixup target "*symbol*"; possible zero length section

This message is issued only for VxD links when the target of a relocation does not have a section. With LINK386 (an older version), an OMF GROUP record (generated by a MASM GROUP directive) may have been used to combine the zero length section with another non-zero length section. COFF format does not support the GROUP directive and zero-length sections. When LINK automatically converts this type of OMF objects to COFF, this error may occur.

Linker Tools Error LNK1189

LIBTOOMANYMEMBERS:: library limit of number objects exceeded

The limit of 65535 objects or members in a library has been exceeded.

[Return to Main Linker Error Page](#)

Linker Errors LNK1190 to LNK1581

Linker Tools Error LNK1190

invalid fixup found, type type

The object file has become corrupted. Recompile.

Linker Tools Error LNK1194

cannot delay-load dll name due to import of data symbol symbol name; relink without /DELAYLOAD:dll name

You cannot delay load a DLL if data is imported from it.

Linker Tools Error LNK1195

target machine "*machine*" requires "*option*"

Add the required option.

Linker Tools Error LNK1196

invalid or corrupt import object: unknown version

The import library has become corrupted. Rebuild the library.

Linker Tools Error LNK1197

invalid or corrupt import object: unknown type

The import library has become corrupted. Rebuild the library.

Linker Tools Error LNK1198

invalid or corrupt import object: unknown name type

The import library has become corrupted. Rebuild the library.

Linker Tools Error LNK1199

invalid or corrupt import object: non-zero reserved fields

The import library has become corrupted. Rebuild the library.

Linker Tools Error LNK1200

error reading program database "*filename*"

The given program database (PDB) could not be read, probably because it is corrupted. If filename is the PDB for an object file, recompile the object file; use the /debug:full and /pdbfile options. If filename is the PDB for the main output file and this error occurred during an incremental link, delete the PDB and relink.

Linker Tools Error LNK1201

error writing to program database "*filename*"; check for insufficient disk space

LINK could not write to the program database (PDB) for the output file. One of the following may be a cause:

- The file is corrupted. Delete filename and relink.
- There was not enough disk space to write to the file.
- The drive being written to was not available, possibly due to a network problem.

Linker Tools Error LNK1202

"filename" missing debugging information for referencing module

The given program database (PDB) for an object file was invalid. Recompile the object file; use CL's /Zi option.

Linker Tools Error LNK1203

"filename" missing current debugging information for referencing module

The given program database (PDB) for an object file was invalid. Recompile the object file; use CL's /Zi option.

Linker Tools Error LNK1204

"filename" compiled /Yc /Yu /Z7; cannot create PDB; recompile with /Zi

Multilayered program databases (PDBs) are not supported in combination with old-style debugging information. Recompile using CL's /Zi option.

Linker Tools Error LNK1206

cannot overwrite Visual C++ 1.0 PDB *"filename"*; delete and rebuild

This version of LINK cannot write to an existing program database (PDB) created using older versions of the visual development environment. Delete filename and rebuild.

Linker Tools Error LNK1207

incompatible PDB format in *"filename"*; delete and rebuild

This version of LINK cannot write to the existing program database (PDB). Delete filename and rebuild.

Linker Tools Error LNK1209

program database *"filename"* differs from previous link; relink or rebuild

The given program database (PDB) is invalid and possibly corrupt. Relink. If filename is also the PDB for an object file, recompile to recreate the PDB.

Linker Tools Error LNK1210

insufficient memory for incremental link; relink with /INCREMENTAL:NO

There was not enough virtual memory available for LINK to create the incremental status (.ILK) file.

Linker Tools Error LNK1211

precompiled type information not found; *"filename"* not linked or overwritten

The given object file, compiled with /Yc, either was not specified in the LINK command or was overwritten.

Linker Tools Error LNK1212

error opening program database; file is in use

The PDB is already in use by another application.

Linker Tools Error LNK1213

unexpected import object encountered

The import library has become corrupted. Rebuild the library.

Linker Tools Error LNK1221

a subsystem can't be inferred and must be defined

The linker does not have enough information to infer which subsystem you will target your application. To fix this error, use the /SUBSYSTEM option.

Linker Tools Error LNK1561

entry point must be defined

The symbol specified by the /ENTRY option is not defined.

Linker Tools Error LNK1581

corrupted object or old compiler (bad Pcode entry point)

Your object file contained a bad entry point. The object file is probably corrupt. See [Corrupt Object File](#) for more information.

[Return to Main Linker Error Page](#)

Linker Errors LNK2001 to LNK2014**Linker Tools Error LNK2001**

unresolved external symbol "*symbol*"

Code will generate this error message if it references something (like a function, variable, or label) that the linker cannot find in all the libraries and object files it searches. In general, there are two reasons this error occurs: what the code asks for doesn't exist (the symbol is spelled incorrectly or uses the wrong case, for example), or the code asks for the wrong thing (you are using mixed versions of the libraries; some from one version of the product, others from another version).

Depending on the calling convention used, an at sign (@) and a decimal number may be appended to the symbol name (see [ATTRIBUTES Properties and Calling Conventions](#)).

Numerous kinds of coding and build errors can cause LNK2001. Several specific causes are listed below, and some have more detailed explanations.

Coding Problems:

- In Fortran programs, some of the causes of the LNK2001 message can be one of the following:
 - Calling a routine with a different number of arguments or the wrong argument types than it was declared. If an argument count mismatch is intentional, then use the Fortran OPTIONAL attribute (an explicit interface is required), or use !DEC\$ ATTRIBUTES C to give the called routine the C attribute (this must be specified both in the calling

- routine and in the called routine). For more information, see the Programmer's Guide section on [mixed-language programming](#).
- Unresolved references to Visual Fortran library routine names can be caused by omitting the appropriate **USE** statement required by that routine. For example, calling the [TRACEBACKQQ](#) routine requires the **USE DFLIB** statement and [DLGINIT](#) requires **USE DFLOGM**. Similarly, a specific **USE** statement is needed to [call IMSL routines](#) and a specific **INCLUDE** statement is needed to [call CXML routines](#).
 - Unresolved references to `_WinMain@16` can be caused by specifying a project type as Fortran Windows for a traditional application (Fortran Windows applications require a WinMain function declaration and interface). In this case, create a new Fortran project with the correct [project type](#) (such as Fortran Console), copy the files to the new project directory, and add them to the new project (see [Defining Your Project](#)).
 - If the routine you are calling is a Win32 API routine, you may need to specify its library in Link tab, Object/Library modules field in the Project Settings dialog box (click Settings in the Project menu). To see which library is required, read the routines description and click on QuickView in the Platform SDK documentation (see [Calling Win32 Routines and the Visual Fortran Windows Module](#)).
 - Mismatched case in your code or module-definition (.DEF) file can cause LNK2001. For example, if you named a variable "var1" in one C++ source file and tried to access it as "VAR1" in another, you would receive this error. The solution is to exactly match the case of the symbol in all references.
 - A project that uses function inlining yet defines the functions in a .CPP file rather than in the header file can cause LNK2001.
 - If you are using C++, make sure to use extern "C" when calling a C function from a C++ program. By using extern "C" you force the use of the C naming convention. Be aware of compiler switches like /Tp or /Tc that force a file to be compiled as a C (/Tc) or C++ (/Tp) file no matter what the filename extension, or you may get different function names than you expect.
 - Attempting to reference functions or data that don't have external linkage causes LNK2001. In C++, inline functions and const data have internal linkage unless explicitly specified as extern.
 - A missing function body or variable will cause LNK2001. Having just a function prototype or extern declaration will allow the compiler to continue without error, but the linker will not be able to resolve your call to an address or reference to a variable because there is no function code or variable space reserved.
 - Name decoration incorporates the parameters of a function into the final decorated function name. Calling a function with parameter types that do not match those in the function declaration may cause LNK2001.
 - Incorrectly included prototypes will cause the compiler to expect a function body that is not provided. If you have both a class and non-class

implementation of a function F, beware of C++ scope-resolution rules.

- When using C++, make sure that you include the implementation of a specific function for a class and not just a prototype in the class definition.
- Attempting to call a pure virtual function from the constructor or destructor of an abstract base class will cause LNK2001 since by definition a pure virtual function has no base class implementation.
- Only global functions and variables are public.

Functions declared with the static modifier by definition have file scope.

Static variables have the same limitation. Trying to access any static variables from outside of the file in which they are declared can result in a compile error or LNK2001.

A variable declared within a function (a local variable) can only be used within the scope of that function.

C++ global constants have static linkage. This is different than C. If you try to use a global constant in C++ in multiple files you get error LNK2001.

One alternative is to include the const initializations in a header file and include that header in your .CPP files when necessary, just as if it was a function prototype. Another alternative is to make the variable non-constant and use a constant reference when accessing it.

Compiling and Linking Problems:

- The names of the run-time libraries needed at link time are included in the object file module by the compiler. If you use the /NOD (/NODEFAULTLIB) option, these libraries will not be linked into the project unless you have explicitly included them. Using /NOD will cause error LNK2001 in this case.
- If you are using Unicode and MFC, you will get an unresolved external on _WinMain@16 if you don't create an entrypoint to wWinMainCRTStartup. Use the /ENTRY option or type this value in the Project Settings dialog box. (To find this option in the development environment, click Settings on the Project menu, then click the Link tab, and click Output in the Category box.)
- See the following Microsoft Knowledge Base articles for more information (either search MSDN if it is installed or use a Web browser to search for the following article Q numbers at <http://support.microsoft.com/support>):
 - Q125750 "PRB: Error LNK2001: '_WinMain@16': Unresolved External Symbol"
 - Q131204 "PRB: Wrong Project Selection Causes LNK2001 on _WinMain@16"
 - Q100639 "Unicode Support in the Microsoft Foundation Class Library"
- Linking code compiled with /MT with the library LIBC.LIB causes LNK2001 on _beginthread, _beginthreadex, _endthread, and _endthreadex.
- When compiling with /MD, a reference to "func" in your source becomes a reference "__imp__func" in the object since all the run-time is now held within a DLL. If you try to link with the static libraries LIBC.LIB or LIBCMT.LIB, you will get LNK2001 on __imp__func. If you try to link with

MSVCxx.LIB when compiling without /MD you will not always get LNK2001, but you will likely have other problems.

- Linking code compiled with an explicit or implicit /ML to the LIBCMT.LIB causes LNK2001 on _errno.
- Linking with the release mode libraries when building a debug version of an application can cause LNK2001. Similarly, using an /Mxd option (/MLd, /MTd, or /MDd) and/or defining _DEBUG and then linking with the release libraries will give you potential unresolved externals (among other problems). Linking a release mode build with the debug libraries will also cause similar problems.
- Mixing versions of Microsoft libraries and compiler products can be problematic. A new compiler version's libraries may contain new symbols that cannot be found in the libraries included with previous versions. Use DUMPBIN to find out if a symbol is in a 32-bit object file or library.
- There is currently no standard for C++ naming between compiler vendors or even between different versions of a compiler. Therefore linking object files compiled with other compilers may not produce the same naming scheme and thus cause error LNK2001.
- Mixing inline and non-inline compile options on different modules can cause LNK2001. If a C++ library is created with function inlining turned on (/Ob1 or /Ob2) but the corresponding header file describing the functions has inlining turned off (no inline keyword), you will get this error. To prevent this problem, have the inline functions defined with inline in the header file you are going to include in other files.
- If you are using the Visual C++ #pragma inline_depth compiler directive, make sure you have a value of 2 or greater set, and make sure you are using the /Ob1 or /Ob2 compiler option.
- Omitting the LINK option /NOENTRY when creating a resource-only DLL will cause LNK2001.
- Using incorrect /SUBSYSTEM or /ENTRY settings can cause LNK2001. For example, if you write a character-based application (a console application) and specify /SUBSYSTEM:WINDOWS, you will get an unresolved external for WinMain. For more information on these options and entry points, see the /SUBSYSTEM and /ENTRY linker options.

Export Problems:

- When you are porting an application from 16 to 32 bits, LNK2001 can occur. The current 32-bit module-definition (.DEF) file syntax requires that __cdecl, __stdcall, and __fastcall functions be listed in the EXPORTS section without underscores (undecorated). This differs from the 16-bit syntax, where they must be listed with underscores (decorated). For more information, see the description of the EXPORTS section of module-definition files.
- Any export listed in the .DEF file and not found will cause LNK2001. This could be because it does not exist, is spelled incorrectly, or uses decorated

names (.DEF files do not take decorated names). This error message is followed by fatal error LNK1120.

The following sections give more detailed information on some of the issues named in the above list.

- Missing function body or variable

Having just a function prototype will allow the compiler to continue without error, but the linker will not be able to resolve your call to an address because there is no function code or variable space reserved. You will not see this error until you create an actual call to the function that the linker must resolve.

- Name decoration

Normally this refers to C++ naming conventions, but it applies to a number of non-C++ cases as well. C++ by default will use the name of a function, its parameters and its return type when calculating a name for a function.

Use extern "C" when calling a C function from a C++ program. Extern "C" forces use of the C naming convention for non-class C++ functions.

Linker Tools Error LNK2003

[gp relative fixup to symbol not in .sdata "module"](#)

The /Gt value specified when compiling an instance of a type differed from the value specified when compiling the reference to the type. Rebuild the object files with consistent values for /Gt. This error is followed by fatal error LNK1165.

Linker Tools Error LNK2004

[gp relative fixup overflow; sdata section \("section"\) is too large](#)

The .sdata section was too large. Rebuild using smaller values for /Gt. This error is followed by fatal error LNK1165.

Linker Tools Error LNK2005

[symbol already defined in object](#)

The given symbol, displayed in its decorated form, was multiply defined. One of the following may be a cause:

- The most common cause of this error is accidentally linking with both the single-threaded and multithreaded libraries, or with both the static and dynamic libraries, or with both debug and non-debug versions. Ensure that the application project includes only the appropriate libraries and that any third-party libraries have appropriately created single-threaded or multithreaded versions, or appropriately created static or dynamic versions.

The types of the Fortran and C libraries must match. For more information, see [Specifying Consistent Library Types](#).

- The given symbol was a packaged function (created by compiling with /Gy) and was included in more than one file but was changed between compilations. Recompile all files that include the symbol.
- The given symbol was defined differently in two member objects in different libraries, and both member objects were used.
- An absolute was defined twice, with a different value in each definition. This error is followed by fatal error LNK1169.

Linker Tools Error LNK2006

TOC relative fixup to symbol not in TOC "*name*"; fixup ignored
LINK found an invalid fixup and ignored it.

Linker Tools Error LNK2007

TOC relative fixup overflow; TOC is too large; fixup ignored
LINK found an invalid fixup and ignored it.

Linker Tools Error LNK2008

Fixup target is not aligned "*alignment*"
LINK found a fixup target in your object file that was not aligned properly.

Linker Tools Error LNK2009

Fixup target must be absolute "*name*" w/o -FIXED; fixup ignored
LINK found an invalid fixup and ignored it.

Linker Tools Error LNK2010

Duplicate IMGLUE relocations for "*name*"
LINK found duplicate relocations for "*name*" in your object file. The object file is probably corrupt. See [Corrupt Object File](#) for more information.

Linker Tools Error LNK2011

precompiled object not linked in; image may not run
If you use precompiled headers, LINK now requires that all of the object files created with precompiled headers must be linked in. If you have a source file that you use to generate a precompiled header for use with other source files, you now must include the object file created along with the precompiled header.

Linker Tools Error LNK2012

No NOP following relocationtype relocation to "*symbol*"
TOCINDIRCALL and TOCCALLREL require a NOP following the instruction that has this type of relocation. This is mainly useful for compiler and assembler developers who use the Microsoft linker.

Linker Tools Error LNK2013

Fixup overflow. Target "*symbol*" is out of range

This error occurs when the executable is too large or the error may indicate a problem with assembly code. To work around the problem:

- Use shared libraries.
- TOC space is required for global data accessed from a module other than where it is defined.

You can eliminate the TOC space by moving such data to the module where it is referenced. However, if data is referenced in multiple modules, then the TOC space cannot be eliminated to save space.

Linker Tools Error LNK2014

TLS relative fixup overflow; *.tls* section ("*section*") is too large

The section created for thread-local storage (*.tls*) is too large. TLS data must fit in 32 KB. This data is created using the `__declspec(thread)` storage class modifier in the declaration and definition of that data. To work around the problem:

- Reduce the amount of thread-local data in the code.
- Use dynamic TLS by calling functions such as `TlsAlloc` and `TlsFree`.

[Return to Main Linker Error Page](#)

Linker Warnings LNK4001 to LNK4039

Linker Tools Warning LNK4001

no object files specified; libraries used

The command did not specify any object files as input. The tool used objects from the libraries that were specified as input.

Linker Tools Warning LNK4002

symbol defined in object

The given symbol, displayed in its decorated form, was specified in its undecorated form in object, but a unique match to a decorated symbol could not be found. This warning is always preceded by warning LNK4022 and followed by fatal error LNK1152.

Linker Tools Warning LNK4003

invalid library format; library ignored

The file header for the input library was invalid. The tool did not use the library.

Linker Tools Warning LNK4005

no objects used from library library

The library was specified in the command, but no references were resolved from that library.

Linker Tools Warning LNK4006

symbol already defined in object; second definition ignored

The given symbol, displayed in its decorated form, was multiply defined. When this warning is encountered, symbol will be added twice, but only its first form will be used. Tips:

- The given symbol may be a packaged function, created by compiling with /Gy. This symbol was included in more than one file but was changed between compilations. Recompile all files that include the symbol.
- The given symbol may have been defined differently in two member objects in different libraries.
- An absolute may have been defined twice, with a different value in each definition.
- If the error message is received when combining libraries, symbol already exists in the library being added to.

Linker Tools Warning LNK4010

invalid version number "*number*"; default version assumed

Either the major or the minor number in the specified version argument was not valid. Each part of a version specification must be an integer in the range 1-65535. Options that take a version argument include /SUBSYSTEM and /VERSION.

Linker Tools Warning LNK4011

unknown subsystem "*argument*"; default assumed

The /SUBSYSTEM option was incorrectly specified. Either no argument was specified or argument did not have the expected syntax. The tool assumed an appropriate default.

Linker Tools Warning LNK4012

invalid value "*argument*", must be keywords; option ignored

The given argument was not a valid keyword for the option it was specified with. The valid arguments are shown in the given list of keywords.

Linker Tools Warning LNK4013

image size imagesize exceeds specified maximum maxsize

The base address file specified with the /BASE option listed the maximum size of the program as maxsize, but maxsize was less than the resulting image size of the program.

Linker Tools Warning LNK4014

cannot find member object "*objectname*"

LIB could not find objectname in the library. The /REMOVE and /EXTRACT options require the full name of the member object that is to be deleted or

copied to a file. The full name includes the path of the original object file. To see the full names of member objects in a library, use DUMPBIN /ARCHIVEMEMBERS or LIB /LIST.

Linker Tools Warning LNK4015

memorytype commit exceeds reserve; commit size set to *size*

The value specified for the commit size for the memorytype exceeded the value specified for the reserve size. The tool assumed size as the value of the commit size.

Linker Tools Warning LNK4017

keyword statement not supported for the target platform; ignored

The given module-definition statement is not supported by this version of the linker tools. The tool ignored the statement in the .DEF file.

Linker Tools Warning LNK4018

too many type indexes in PDB "*filename*", discarding subsequent type information

There are more than 64K type indexes in the program database (PDB). The PDB might contain unneeded type information. Delete the existing PDB and rebuild. If the warning persists, then compile some modules without debugging information.

Linker Tools Warning LNK4019

corrupt string table (table end); new end assumed

The object file is corrupt. The tool attempted to correct the problem. To prevent this warning, rebuild the object file. See [Corrupt Object File](#) for more information.

Linker Tools Warning LNK4022

cannot find unique match for symbol "*symbol*"

LINK or LIB found multiple matches for the given undecorated symbol and it could not resolve the ambiguity. No output file (.EXE, .DLL, .EXP, or .LIB) is produced. This warning is followed by one warning LNK4006 for each duplicate symbol and is finally followed by fatal error LNK1152. To prevent this warning, specify the symbol in its decorated form. Run DUMPBIN on the object to see decorated names.

Linker Tools Warning LNK4023

base address address not a 64K multiple; assumed newaddress

The base address specified with the /BASE option must be a multiple of 64K. The tool assumed a newaddress value for the base address.

Linker Tools Warning LNK4027

CVPACK error

LINK called CVPACK to process the debugging information, but one of the

following occurred:

- The call failed. Be sure that CVPACK is in a directory specified in the PATH environment variable and that there is sufficient memory.
- CVPACK issued an error. Errors generated by CVPACK have the form CKnnnn.

Linker Tools Warning LNK4028

option option is obsolete; ignored

The given option is not supported. LINK ignored the option and continued the build.

Linker Tools Warning LNK4030

invalid file offset number for category; *category* not dumped

DUMPBIN could not display information for the given category. Probably the file is invalid or corrupt. See [Corrupt Object File](#) for more information.

Linker Tools Warning LNK4031

no subsystem specified; CONSOLE assumed

The /SUBSYSTEM option was not specified, and LINK was not able to determine a subsystem from the input files.

Linker Tools Warning LNK4033

converting object format from OMF to COFF

The format of the object file is OMF. This tool requires COFF input. To permanently convert the file, run EDITBIN on the file with no options.

Linker Tools Warning LNK4037

"symbol" does not exist; ignored

The symbol could not be ordered using the /ORDER option because it could not be found in the program. Check the specification of symbol in the ordering file. Note: LINK cannot order static functions because static function names are not public symbol names. When /ORDER is specified, this linker warning is generated for each symbol, in the order file, that is either static or not found.

Linker Tools Warning LNK4038

displaying raw data in number units per line

The /RAWDATA option was specified without a formatting argument. DUMPBIN assumed a format value of number units per line of the specified or assumed data type, as follows:

Data Type	Number
none	16
BYTES	16

SHORTS 8
LONGS 4

Linker Tools Warning LNK4039

section "*name*" specified with /SECTION option does not exist

DUMPBIN or EDITBIN could not find a section called *name* in the input file. To see the sections in a file, run DUMPBIN with the /HEADERS option.

[Return to Main Linker Error Page](#)

Linker Warnings LNK4040 to LNK4069

Linker Tools Warning LNK4040

corrupt string table (size); string table ignored

The string table in the object file is corrupt. The program may not run as expected. Rebuild the object file. See [Corrupt Object File](#) for more information.

Linker Tools Warning LNK4041

no edit options specified

EDITBIN was run on the input file, but no options were used. If the file format was not COFF, EDITBIN attempted to convert it. Otherwise, EDITBIN took no action on the file.

Linker Tools Warning LNK4042

object specified more than once; extras ignored

The same object file was specified more than once in the LINK or LIB command. The result is the same as if the file were specified only the first time.

Linker Tools Warning LNK4043

invalid /ALIGN value number (must be power of 2); assumed default

The argument specified with the /ALIGN option was not a power of 2.

Linker Tools Warning LNK4044

unrecognized option "*option*"; ignored

The given option is not a valid option for this tool. The tool ignored the option. To see a list of valid options, run the tool with no arguments or options.

Linker Tools Warning LNK4045

creating LINK_REPRO test case in dir "*directory*"

The LINK_REPRO environment variable, when set, specifies a directory where the linker will copy object files, libraries, and other files. Microsoft can use these files to recreate internal linker errors. The linker will only copy files if LINK_REPRO is set. If this is not what you want to do, remove the LINK_REPRO environment variable by typing the following: `set LINK_REPRO=`

Linker Tools Warning LNK4046

LINK_REPRO is set to current directory; ignored

The LINK_REPRO environment variable should not be set to the current directory. Create a new directory, set LINK_REPRO to the new directory, and rerun the linker.

Linker Tools Warning LNK4047

cannot edit library file "*filename*"; ignored

EDITBIN cannot process a library file.

Linker Tools Warning LNK4048

Invalid format file; ignored

EDITBIN could not process the file because it was not a valid object file, executable file, or dynamic-link library (DLL).

Linker Tools Warning LNK4049

exported symbol "*symbol*" also imported

The symbol was both exported from and imported to the program. This warning is displayed when the LINK warning level (set by the /WARN option) is greater than 1.

Linker Tools Warning LNK4051

extra delimiter in /EXPORT specification

The specification of the /EXPORT option ended with a comma (,).

Linker Tools Warning LNK4052

using .EXP file; ignoring .DEF file "*filename*"

The .DEF file specified in the LINK command was ignored because an .EXP file was also specified. The .DEF file was already processed when the .EXP file was created and is not necessary in this step of the build process.

Linker Tools Warning LNK4055

invalid fixup found, address *address*; fixup ignored

The object contained an invalid fixup. The resulting executable file will probably not run correctly. Rebuild the object file. See [Corrupt Object File](#) for more information.

Linker Tools Warning LNK4056

extra arguments ignored for option "*option*"

The option was specified with too many arguments. The additional arguments were ignored.

Linker Tools Warning LNK4057

"*library1*" should precede "*library2*" in the link command line

Check the order of your libraries. The developer of *library1* determined that it should be searched by the linker before *library2*. For example, *library1* might

contain an updated version of a function in *library2*. If you specify *library2* first, the linker will link in the old version rather than the newer version from *library1*.

Linker Tools Warning LNK4059

resourcefile already specified; additional resource file ignored

Only one resource file is allowed. The first .RES file specified in the command was used.

Linker Tools Warning LNK4060

stub file missing full MS-DOS header; rebuild stub with /KNOWEAS 16-bit LINK option

The MS-DOS application specified with the /STUB option does not have a full 40-byte header. The stub program may not run as expected. To rebuild the 16-bit stub with a full 40-byte header, use the undocumented /KNOWEAS option with the 16-bit LINK version 5.5x or higher.

Linker Tools Warning LNK4061

/NOSTUB ignored because no DOS stub is present

EDITBIN could not remove the stub file because it could not find one in your executable file.

Linker Tools Warning LNK4062

"*option*" not compatible with "*platform*" target machine; option ignored

The given *option* is not supported for the target *platform* of this build. LINK ignored the option and continued the build.

Linker Tools Warning LNK4064

conflicting subsystem; image may not run

You tried to link in a library whose subsystem differs from your executable's subsystem value, and your executable file may not run. For more information on subsystems, see the /SUBSYSTEM linker option. Note: You will not receive this warning if one subsystem is CONSOLE and the other's subsystem is WINDOWS.

Linker Tools Warning LNK4065

"*function*" cannot be ordered; ignored

The given function was not compiled as a packaged function. Recompile using /Gy.

Linker Tools Warning LNK4066

DLL contains .sdata section

One or more objects was compiled using /Gt. Objects compiled with /Gt cannot be used to create a DLL.

Linker Tools Warning LNK4067

ambiguous entry point; selected "*entry*"

LINK found multiple entry points for the program. It assumed the given entry.

Linker Tools Warning LNK4068

/MACHINE not specified; defaulting to machinetype

LINK did not find a machine specification. It assumed the given machine type. If the default is incorrect, resulting in fatal error LNK1112, relink using the /MACHINE option.

Linker Tools Warning LNK4069

cannot create map and/or checksum image; checksum set to 0

This warning can occur if IMAGEHLP.DLL is not available.

[Return to Main Linker Error Page](#)

Linker Warnings LNK4070 to LNK4099**Linker Tools Warning LNK4070**

/OUT:filename directive in .EXP differs from output filename "*filename*"; ignoring directive

The filename specified in the NAME or LIBRARY statement when the .EXP file was created differs from the output filename that was either assumed by default or specified with the /OUT option. A client program that uses the resulting DLL might encounter problems.

Linker Tools Warning LNK4071

cannot be incrementally linked on subsequent links

LINK found multiple definitions for one or more symbols, but /FORCE or /FORCE:MULTIPLE was used to create an output file regardless of errors. LINK deleted the incremental status (.ILK) file.

Linker Tools Warning LNK4072

section count count exceeds max (*number*); image may not run

The number of sections (a COFF .EXE or .DLL file is composed of sections, similar to segments) in the output file is greater than the limit that can be handled by the loader.

You can view the names of the sections in your executable file by using the DUMPBIN program with the default /SUMMARY option. See the DUMPBIN Reference for more information. To reduce the number of sections in your executable file:

- Remove unneeded data_seg, code_seg or alloc_text pragmas. See data_seg, code_seg, and alloc_text.
- Use the /MERGE option with LINK to combine sections. See /MERGE.

Linker Tools Warning LNK4073

cannot create map for .ILK file; linking nonincrementally

There was not a large enough contiguous space in shared memory for LINK to create the incremental status (.ILK) file. LINK performed a nonincremental build.

Linker Tools Warning LNK4074

unable to load filename; cannot checksum

The linker tried to load the dynamic-link library filename but could not. A function in filename is used to checksum the executable file. One of the following may be a cause:

- The linker could not find filename because it is not in the current directory or in a directory in the PATH environment variable.
- There was a problem with filename. Recopy filename from the installation disks.

Linker Tools Warning LNK4075

ignoring "*option1*" due to "*option2*" specification

The *option2* specification overrides *option1*.

Linker Tools Warning LNK4076

invalid incremental status file "*filename*"; linking nonincrementally

LINK cannot write to the incremental status (.ILK) file. Either filename is corrupt or it is not an incremental linking database. Remove the file and relink.

Linker Tools Warning LNK4077

exports file "*filename*" used; ignoring other export specifications

The given exports file provides the program's exports. Exports specified by the /EXPORT option or the EXPORTS statement in a .DEF file were ignored.

Linker Tools Warning LNK4078

multiple "*section*" sections found with different attributes

LINK found two or more sections that have the same name but different attributes. Probably an import library or exports file was created by a previous version of LINK or LIB. Recreate the file and relink.

Linker Tools Warning LNK4079

no write permission for incremental status file "*filename*"; linking nonincrementally

LINK cannot write to the incremental status (.ILK) file because filename does not have the correct permissions.

Linker Tools Warning LNK4081

insufficient disk space; linking nonincrementally

There is not enough space on disk for LINK to create or write to the incremental status (.ILK) file.

Linker Tools Warning LNK4082

unable to find entrypoint function in module; cannot checksum
The function CheckSumMappedFile() does not exist in IMAGEHLP.DLL.

Linker Tools Warning LNK4083

module "*name*" does not exist; ignored
The module "*name*" you gave in your order file does not exist. Check the spelling. For more information on order files, see the /ORDER linker option.

Linker Tools Warning LNK4084

total image size size exceeds max (256MB); image may not run
The application exceeds the limit of 256 megabytes.

Linker Tools Warning LNK4085

more than number comments; skipping exestr
This warning occurs when the linker tool is converting an OMF object to a COFF object. It indicates there are more than 256 comments in the OMF module.

Linker Tools Warning LNK4086

entrypoint '*function*' is not __stdcall with 12 bytes of arguments; image may not run
The entry point for a DLL must be __stdcall. Either recompile the function with the /Gz option or specify __stdcall or WINAPI when you define the function.

Linker Tools Warning LNK4087

CONSTANT keyword is obsolete; use DATA
This warning occurs if you use CONSTANT with your exports in a module definition file or on the command line. Use DATA instead.

Linker Tools Warning LNK4088

image being generated due to /FORCE option; image may not run
The linker encountered an error. Normally it deletes the image it created because the image may not run due to the error. Since you used the /FORCE option, however, the linker did not remove the image.

Linker Tools Warning LNK4089

all references to "dynamic-link library" discarded by /OPT:REF
The linker discarded all packaged functions that referenced exports in dynamic-link library. As a result, dynamic-link library and its import library are unneeded.

Linker Tools Warning LNK4090

No IFGLUE relocation in module with branch to glue code "*name*"
Your object file contained bad relocation information. Recopy or recreate the object file. See [Corrupt Object File](#) for more information.

Linker Tools Warning LNK4092

shared section "*section*" contains relocations; image may not run correctly
The linker emits this warning whenever you have a shared section to warn you

of a potentially serious problem. One way to share data between multiple processes is to mark a section as "shared." Marking a section as shared can cause problems, however.

Linker Tools Warning LNK4093

Drive/Directory component ignored in "*module-definition*" statement

Your module-definition (.DEF) file used a statement incorrectly. The LIBRARY or NAME statements cannot contain directory or drive information.

Linker Tools Warning LNK4094

filename is an MS-DOS executable; use EXEHDR to dump it

The DUMPBIN program only works with COFF-format executable files. Use EXEHDR on MS-DOS executables. EXEHDR is included with Visual C++ 1.5.

Linker Tools Warning LNK4095

filename is an NE format executable; use EXEHDR to dump it

The DUMPBIN program only works with COFF-format executable files. Use EXEHDR on NE format executables. EXEHDR is included with Visual C++ 1.5.

Linker Tools Warning LNK4096

/BASE value "*number*" is invalid for Windows 95; image may not run

The base address you specified is invalid. Windows 95 executable files must have a base address greater than 0x400000. For more information on base addresses, see the /BASE linker option.

Linker Tools Warning LNK4097

duplicate name "*name*" in order file; using last one

Order files specify the order you want the linker to load your functions. In your order file, a function appears more than once, so the linker ignored all references to it except the last. For more information on order files, see the /ORDER linker option.

Linker Tools Warning LNK4098

defaultlib "*library*" conflicts with use of other libs; use /NODEFAULTLIB:library

You are trying to link with incompatible libraries.

Note: The run-time libraries now contain directives to prevent mixing different types. You'll receive this warning if you try to use different types or debug and non-debug versions of the run-time library in the same program. For example, if you compiled one file to use one kind of run-time library and another file to use another kind (for example, single-threaded versus multithreaded) and tried to link them, you will get this warning.

The types of the Fortran and C libraries must match. For more information, see [Specifying Consistent Library Types](#).

You can use the linker's /VERBOSE:LIB switch to determine which libraries the linker is searching. If you receive LNK4098 and want to create an executable file

that uses, for example, the single-threaded, non-debug run-time libraries, use the `/VERBOSE:LIB` option to find out which libraries the linker is searching. The linker should print `LIBC.LIB` and not `LIBCMT.LIB`, `MSVCRT.LIB`, `LIBCD.LIB`, `LIBCMTD.LIB`, or `MSVCRTD.LIB` as the libraries searched. You can tell the linker to ignore the the incorrect run-time libraries by typing the incorrect libraries in the Ignore Libraries text box on the Link tab of the Settings dialog box in the visual development environment or by using the `/NODEFAULTLIB:library` option with `LINK` for each library you want to ignore. See the `/NODEFAULTLIB` linker option for more information.

The table below shows which Visual C++ libraries should be ignored depending on which run-time library you want to use.

To use this run-time library	Ignore these libraries
Single-threaded (<code>libc.lib</code>)	<code>libcmt.lib</code> , <code>msvcrt.lib</code> , <code>libcd.lib</code> , <code>libcmt.d.lib</code> , <code>msvcrt.d.lib</code>
Multithreaded (<code>libcmt.lib</code>)	<code>libc.lib</code> , <code>msvcrt.lib</code> , <code>libcd.lib</code> , <code>libcmt.d.lib</code> , <code>msvcrt.d.lib</code>
Multithreaded using DLL (<code>msvcrt.lib</code>)	<code>libc.lib</code> , <code>libcmt.lib</code> , <code>libcd.lib</code> , <code>libcmt.d.lib</code> , <code>msvcrt.d.lib</code>
Debug Single-threaded (<code>libcd.lib</code>)	<code>libc.lib</code> , <code>libcmt.lib</code> , <code>msvcrt.lib</code> , <code>libcmt.d.lib</code> , <code>msvcrt.d.lib</code>
Debug Multithreaded (<code>libcmt.d.lib</code>)	<code>libc.lib</code> , <code>libcmt.lib</code> , <code>msvcrt.lib</code> , <code>libcd.lib</code> , <code>msvcrt.d.lib</code>
Debug Multithreaded using DLL (<code>msvcrt.d.lib</code>)	<code>libc.lib</code> , <code>libcmt.lib</code> , <code>msvcrt.lib</code> , <code>libcd.lib</code> , <code>libcmt.d.lib</code>

For example, if you received this warning and you want to create an executable file that uses the non-debug, single-threaded version of the run-time libraries, you could use the following options with the

linker: `/NODEFAULTLIB:libcmt.lib /NODEFAULTLIB:msvcrt.lib /NODEFAULTLIB:libcd.lib`

Linker Tools Warning LNK4099

PDB "*filename*" was not found with "*object/library*" or at "*path*"; linking object as if no debug info

The linker was unable to find your .PDB file. Copy it into the directory that contains *object/library*.

[Return to Main Linker Error Page](#)

Linker Warnings LNK4102 to LNK4549

Linker Tools Warning LNK4102

export of deleting destructor *symbol name*; image may not run correctly

The program has attempted to export a deleting destructor. The resulting delete may occur across a DLL boundary such that a process can free memory that it does not own. Make sure that the given symbol is not listed in your .DEF file, and that the symbol is not listed as an argument of the /IMPORT or /EXPORT option in the linker command line.

Linker Tools Warning LNK4103

import of deleting destructor "*symbol name*"; image may not run correctly

The program has attempted to import a deleting destructor. The resulting delete may occur across a DLL boundary such that a process can free memory that it does not own. Make sure that the given symbol is not listed in your .DEF file, and that the symbol is not listed as an argument of the /IMPORT or /EXPORT option in the linker command line.

Linker Tools Warning LNK4104

export of symbol "*symbol*" should be PRIVATE

The *symbol* can be one of the following:

- DllRegisterServer
- DllRegisterServerEx
- DllUnregisterServer
- DllGetClassObject
- DllCanUnloadNow

This warning is emitted when you are building an import library for a DLL and export one of the above functions without specifying it as PRIVATE in the module-definition file. In general these functions are exported for use only by OLE. Placing them in the import library can lead to unusual behavior when a program linked to the library incorrectly makes calls to them.

Linker Tools Warning LNK4105

no argument specified with option *option*; ignoring switch

This warning occurs only when the /LIBPATH option is set. If no directory is specified with this option, then the linker ignores this option and generates this warning message. If you do not need to override the existing environmental library settings, remove the /LIBPATH option from the linker command line. If you want to use an alternate search path for libraries, specify the alternate path following the /LIBPATH option. For example: `link /libpath:c:\foo\lib bar.obj` would direct the linker to search for the required libraries in `c:\foo\lib` before searching in the default locations.

Linker Tools Warning LNK4106

"*section name*" section discarded due to "*option*" specification

The *option* caused the linker to discard the section.

Linker Tools Warning LNK4107

/option not supported by this version of MSPDBxx.DLL; ignored
MSPDBxx.DLL does not support the specified *option*.

Linker Tools Warning LNK4108

/ALIGN specified without */DRIVER* or */VXD*; image may not run
The */ALIGN* option has been specified without also specifying either */DRIVER* or */VXD*. Bad alignment can prevent the operating system from loading the final image. Do not use the */ALIGN* option unless building a driver or vxd.

Linker Tools Warning LNK4194

/DELAYLOAD:dll name ignored
The linker cannot delay load the requested DLL.

Linker Tools Warning LNK4195

unable to load dll name
The linker could not load the requested DLL.

Linker Tools Warning LNK4197

export "*exportname*" specified multiple times; using first specification
An export is specified in multiple, different ways. The linker uses the first specification and ignores the rest. Perhaps the same export is specified both on the command line (via *export:*) and in the *.def* file or listed twice in the *.def* file with different attributes. If an export is specified exactly the same way multiple times, the linker will not issue a warning.

Linker Tools Warning LNK4198

base key "*keyname*" not found - using default
The *keyname*, specified in a */BASE* option, does not appear in the base address file.

Linker Tools Warning LNK4199

/DELAYLOAD: *dllname* ignored; no imports found from *dllname*
The linker ignores *dllname* because it does not need any of the functions that *dllname* exports.

Linker Tools Warning LNK4200

corrupt line number information in object file; ignored
The line number information in the object file has become corrupted. Rebuild.

Linker Tools Warning LNK4543

Section "*section*" with invalid attributes ignored
The linker could not determine the type of section and ignored it. Valid types for sections include text, data, bss, and rdata.

Linker Tools Warning LNK4544

Symbol "*symbol*" specified with `/IMPORT` not referenced in application - ignored
You specified a symbol with the `/IMPORT` option but it was not referenced in your program so the linker ignored it.

Linker Tools Warning LNK4545

Container "*container*" specified with `/IMPORT` not referenced in application - ignored

You specified an import library (container) with the `/IMPORT` option, but your program did not reference the library. The linker ignored the container.

Linker Tools Warning LNK4546

Mismatch in versions detected. Container may not load. Use `/VERBOSE` for details.

The linker detected a problem with versions in your import libraries (containers). To find out which container (and its version number) is causing the problem, rerun LINK with the `/VERBOSE` option.

Linker Tools Warning LNK4547

Setting "*versiontype*" to version

The linker set *versiontype* to version. Possible values for *versiontype* are CURRENTVER, OLDCodeVER, and OLDAPIVER.

Linker Tools Warning LNK4548

Ignoring "*versiontype*" version

The linker ignored version information of *versiontype*. The possible values of *versiontype* are CURRENTVER, OLDCodeVER, and OLDAPIVER.

Linker Tools Warning LNK4549

shared library "*library*" does not have any exports

The shared library was ignored since it contained no exports.

[Return to Main Linker Error Page](#)

Linker Warnings LNK6001 to LNK6026

Linker Tools Warning LNK6001

insufficient disk space; performing full link

There was not enough space on disk for LINK to incrementally link the program. LINK attempted to correct the problem and recover disk space by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6002

invalid .ILK file found; performing full link

The incremental status (.ILK) file was not valid. Probably the previous incremental linking session was interrupted. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6003

LINK options changed; performing full link

The options specified to the linker differed from the options used during the previous incremental linking session. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6004

filename not found or not built by the last incremental link; performing full link

The given file was either deleted or changed since the last incremental linking session. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6005

exports changed; performing full link

One or more exports were added or deleted since the last incremental linking session. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6006

program database pdbname missing; performing full link

LINK could not find the program database (.PDB) file that it created during the last incremental linking session. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6008

object file added; performing full link

An object file was added since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6009

object file removed; performing full link

An object file was removed since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6010

library changed; performing full link

A library was modified since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6011

internal limit exceeded; performing full link

An internal limit related to incremental linking was exceeded. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file. One cause of this warning is an .ILK file that exceeds 8MB. If this size is due to the accumulation of dead space, that space will be recovered in a full link.

Linker Tools Warning LNK6012

precompiled types found; performing full link

An object file was compiled with /Z7 and the program used a precompiled header. LINK attempted to incrementally link by running a full link to recreate the incremental status (.ILK) file. Rebuild the object file with /Zi.

Linker Tools Warning LNK6013

padding exhausted; performing full link

An internal limit related to incremental linking was exceeded. LINK attempted to correct the problem by running a full link to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6014

symbol reference set changed; performing full link

A reference to a symbol in a library was added since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6015

bss (uninitialized data) change; performing full link

A bss symbol was changed since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6016

absolute symbol changed; performing full link

The address of an absolute symbol was changed since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6017

set of library modules referenced changed; performing full link

A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6018

multiple definitions found for "*symbol*"; performing full link

A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6019

directive changed; performing full link

A directive was added or deleted since the last incremental linking session. A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6020

ILK file not built by this linker; performing full link

The version number of the .ILK file was incorrect for this version of LINK. LINK attempted to correct the problem by running a full link to recreate the

incremental status (.ILK) file.

Linker Tools Warning LNK6021

performing full link

A full link was required in order to create the incremental status (.ILK) file.

Linker Tools Warning LNK6022

unable to load .ILK file; performing full link

The .incremental status (.ILK) file is unavailable because of the actions of another process.

Linker Tools Warning LNK6023

unable to extend map for .ILK file; performing full link

The .incremental status (.ILK) file cannot be written to because of the actions of another process.

Linker Tools Warning LNK6024

Mac Resource File "*filename*" changed; performing full link

A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6025

unable to do comdat selection; performing full link

A full link was required in order to recreate the incremental status (.ILK) file.

Linker Tools Warning LNK6026

obsolete DBI format found; performing full link

A full link was required. The format of debugging information from Visual C++ version 4.0 or earlier. Code that was initially linked with version 4.0 and that is now linked with a later version thus has inconsistent debugging information. Therefore, a full link was performed to rectify the inconsistency.

[Return to Main Linker Error Page](#)

BSCMAKE Messages BK1500 through BK4503

The following descriptions cover BSCMAKE Error BK1500 through Warning BK4503:

BSCMAKE Error BK1500

UNKNOWN ERROR

BSCMAKE detected an unknown error condition. Note the circumstances of the error, try to isolate the problem and create a reproducible test case, then report the circumstances to technical support.

BSCMAKE Error BK1501

unknown character '*character*' in option '*option*'

BSCMAKE did not recognize the given character specified for the given option.

BSCMAKE Error BK1502

incomplete specification for option '*option*'

The given option did not contain the correct syntax.

BSCMAKE Error BK1503

cannot write to file '*filename*' [: *reason*]

BSCMAKE could not write to the given file. This message may be followed by a reason from the operating system, giving additional information. One of the following may have occurred:

- The disk was full.
- A hardware error occurred.

BSCMAKE Error BK1504

cannot position in file '*filename*' [: *reason*]

BSCMAKE could not move to a location in the given file. This message may be followed by a reason from the operating system, giving additional information. One of the following may have occurred:

- The disk was full.
- A hardware error occurred.
- The file was truncated. Truncation can occur if the compiler runs out of disk space or is interrupted when it is creating the .SBR file.

BSCMAKE Error BK1505

cannot read from file '*filename*' [: *reason*]

BSCMAKE could not read from the given file. This message may be followed by a reason from the operating system, giving additional information. One of the following may have occurred:

- The file was corrupt.

- The file was truncated. Truncation can occur if the compiler runs out of disk space or is interrupted when it is creating the .SBR file.

BSCMAKE Error BK1506

cannot open file '*filename*' [: *reason*]

BSCMAKE could not open the given file. This message may be followed by a message from the operating system, giving additional information. One of the following may have occurred:

- The file was locked by another process. If reason was Permission denied, the browser in the development environment might be using the file. Close the Browse window and repeat the build.
- The disk was full.
- A hardware error occurred.
- The specified output file had the same name as an existing subdirectory.

BSCMAKE Error BK1507

cannot open temporary file '*filename*' [: *reason*]

BSCMAKE could not open one of its temporary files. This message may be followed by a reason from the operating system, giving additional information. One of the following may have occurred:

- The TMP environment variable was not set to a valid drive and directory.
- The disk was full.

BSCMAKE Error BK1508

cannot delete temporary file '*filename*' [: *reason*]

BSCMAKE could not delete one of its temporary files. This message may be followed by a reason from the operating system, giving additional information. One of the following may have occurred:

- Another process had the file open.
- A hardware error occurred.

BSCMAKE Error BK1509

out of heap space

BSCMAKE ran out of memory, including virtual memory. One of the following may be a solution:

- Free some space on disk.
- Increase the size of the Windows swap file. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory, reboot, and retry.
- Reduce the memory that BSCMAKE will require by using one or more options. Use /Ei or /Es to eliminate some input files. Use /Em to eliminate macro bodies.

BSCMAKE Error BK1510

corrupt .SBR file '*filename*'

The given .SBR file is corrupt or does not have the expected format. Recompile to regenerate the .SBR file.

BSCMAKE Error BK1511

invalid response file specification

BSCMAKE did not understand the command-line specification for the response file. The specification was probably wrong or incomplete. For example, the following specification causes this error: `bscmake @`

BSCMAKE Error BK1512

filename: capacity exceeded

BSCMAKE could not build a browse information file because the number of definitions, references, modules, or other information exceeded the limit. One of the following may be a solution:

- Exclude some information using the /Em, /Es, or /Ei option.
- Omit the /lu option, if it was used.

BSCMAKE Error BK1513

nonincremental update requires all .SBR files

An attempt was made to build a new browse information (.BSC) file, but one or more of the specified .SBR files was truncated. This message is always preceded by warning BK4502, which gives the name of the .SBR file that caused the error.

BSCMAKE can process a truncated, or zero-length, .SBR file only when a .BSC file already exists and is being incrementally updated. One of the following may be a cause:

- The .BSC file was missing.
- The wrong filename was specified.
- The file was corrupted, and a full build was required.

BSCMAKE Error BK1514

all .SBR files truncated, none found in filename

None of the .SBR files specified for an update was a part of the original browse information (.BSC) file. This message is always preceded by warning BK4502, which gives the name of the .SBR file that caused the error. One of the following may be a cause:

- The wrong filename was specified.
- The file was corrupted, and a full build was required.

BSCMAKE Error BK1515

'*bscfile*': incompatible version: cannot incrementally update

The given browse information (.BSC) file was not created with this version of BSCMAKE. A .BSC file can be incrementally built only by the same version of BSCMAKE as the one used to fully build that file.

BSCMAKE Error BK1516

'bscfile' corrupt; cannot incrementally update

The given browse information (.BSC) file was corrupted, possibly due to a system failure during the build. Delete the .BSC file, rebuild all .SBR files, then rebuild the .BSC file.

BSCMAKE Error BK1517

source file for *'filename'* compiled with both */Yc* and */Yu*

The given .SBR file refers to itself. Probably the file was recompiled with the */Yu* option after compiling with */Yc*. Reset the compiler options for the source file to */Yc*, then rebuild all to generate new .SBR files for the project. Do not subsequently compile the same source file using */Yu*.

BSCMAKE Warning BK4500

UNKNOWN WARNING

An unknown warning condition was detected by BSCMAKE. Note the circumstances of the warning, try to isolate the problem and create a reproducible test case, then report it to technical support.

BSCMAKE Warning BK4501

ignoring unknown option *'option'*

BSCMAKE did not recognize the given option and ignored it.

BSCMAKE Warning BK4502

truncated .SBR file *'filename'* not in filename

The given zero-length .SBR file, specified during an update, was not originally part of the browse information (.BSC) file. If a zero-length file that is not part of the original build of the .BSC file is specified during a rebuild of that file, BSCMAKE issues this warning. One of the following may be a cause:

- The wrong filename was specified.
- The file was deleted. (Error BK1513 will result.)
- The file was corrupted, requiring a full build.

BSCMAKE Warning BK4503

minor error in .SBR file *'filename'* ignored

The given .SBR file contained an error that did not halt the build. However, the resulting .BSC file may not be correct. Recompile to regenerate the .SBR file.

C Run-Time Errors R6002 through R6025

The following descriptions cover C Run-Time Errors R6002 through R6025 for mixed-language programming:

C Run-Time Error R6002

floating-point support not loaded

The program needs the floating-point library, but the library was not linked to the program. One of the following may have occurred:

- The program was compiled or linked with an option (such as /FPi87) that required a coprocessor, but the program was run on a machine that did not have a coprocessor installed.
- A format string for a printf or scanf function contained a floating-point format specification, and the program did not contain any floating-point values or variables.
- The compiler minimizes a program's size by loading floating-point support only when necessary. The compiler cannot detect floating-point format specifications in format strings, so it does not load the necessary floating-point routines.
- Use a floating-point argument to correspond to the floating-point format specification, or perform a floating-point assignment elsewhere in the program. This causes floating-point support to be loaded.
- In a mixed-language program, a C library was specified before a FORTRAN library when the program was linked. Relink and specify the C library last.

C Run-Time Error R6005

not enough memory on _exec

Not enough memory was available to load the process being spawned. This error occurs when a child process that was spawned by one of the _exec library routines fails and the operating system cannot return control to the parent process.

C Run-Time Error R6006

invalid format on _exec

The file to be executed by one of the _exec functions was not in the correct format for an executable file. This error occurs when a child process that was spawned by one of the _exec library routines fails and the operating system cannot return control to the parent process.

C Run-Time Error R6007

invalid environment on _exec

During a call to an _exec function, the operating system found that the child process was given an invalid environment block. This error occurs when a child process that was spawned by one of the _exec library routines fails and the

operating system cannot return control to the parent process.

C Run-Time Error R6008

not enough space for arguments

There was enough memory to load the program but not enough memory to create the argv array. One of the following may be a solution:

- Increase the amount of virtual memory available to the program. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory, reboot, and retry.
- Reduce the number and size of command-line arguments.
- Reduce the environment size by removing unnecessary variables.

C Run-Time Error R6009

not enough space for environment

There was enough memory to load the program but not enough memory to create the envp array. One of the following may be a solution:

- Increase the amount of memory available to the program.
- Reduce the number and size of command-line arguments.
- Reduce the environment size by removing unnecessary variables.

If your program uses the compact, large, or huge memory model, this error may be avoided by using LINK's /CPARM:1 command-line option. This option causes unused near heap space to be allocated to the far heap.

C Run-Time Error R6016

not enough space for thread data

The program did not receive enough memory from the operating system to complete a `_beginthread` call. When a new thread is started, the library must create an internal database for the thread. If the database cannot be expanded with memory provided by the operating system, the thread will not begin and the calling process will stop.

C Run-Time Error R6017

unexpected multithread lock error

The process received an unexpected error while trying to access a C run-time multithread lock on a system resource. This error usually occurs if the process inadvertently alters the run-time heap data. However, it can also be caused by an internal error in the run-time or operating-system code.

C Run-Time Error R6018

unexpected heap error

The program encountered an unexpected error while performing a memory-management operation. This error usually occurs if the program inadvertently alters the run-time heap data. However, it can also be caused by an internal

error in the run-time or operating-system code. If your compiler provides a library containing `_heapchk` and `_heapwalk`, you can use these functions to diagnose this error.

C Run-Time Error R6019

unable to open console device

The program called a console function declared in `CONIO.H`, but the operating system did not grant access to the console.

C Run-Time Error R6021

no main procedure

The program does not have a procedure called `main`. Make sure that all object and library modules have been linked into the executable.

C Run-Time Error R6024

not enough space for `_onexit/atexit` table

There was no memory available for the `_onexit` or `atexit` function. This error is caused by a low-memory condition.

C Run-Time Error R6025

pure virtual function call

No object has been instantiated to handle the pure virtual function call. This error is caused by calling a virtual function in an abstract base class through a pointer that is created by a cast to the type of the derived class, but is actually a pointer to the base class. This can occur when casting from a `void*` to a pointer to a class when the `void*` was created during the construction of the base class.

CVPACK Messages CK1001 through CK4010

The following descriptions cover CVPACK error CK1001 through warning CK4010:

CVPACK Fatal Error CK1001

out of memory

CVPACK ran out of memory. One of the following may be a solution:

- Free some space on disk.
- Increase the virtual memory available. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory, reboot, and retry.
- Recompile one or more of the object files without debugging information. If the file was compiled using [/debug:full](#) (or /Zi or /Z7), either use /debug:minimal (or /Zd) or do not use a debugging option.

CVPACK Fatal Error CK1002

out of memory

CVPACK ran out of memory. One of the following may be a solution:

- Free some space on disk.
- Increase the size of the Windows NT (including Windows 2000) paging file, or the Windows swap file.
- Recompile one or more of the object files without debugging information. If the file was compiled using /Zi or /Z7, either use /Zd or do not use a debugging option.

CVPACK Fatal Error CK1003

cannot open file : *filename*

CVPACK could not open the given file. One of the following may be a cause:

- The file does not exist. Check the spelling of the filename and path.
- The file was opened or deleted by another process.
- There was not enough space on disk.

CVPACK Fatal Error CK1004

filename is read-only

The given file is a read-only file, or the path is on a read-only drive.

CVPACK Fatal Error CK1005

invalid executable file

CVPACK could not process the executable file. One of the following may be a cause:

- The debugging information in the executable file is corrupt.

- The executable file is a zero-length file.

CVPACK Fatal Error CK1006

invalid module *module*

The module does not have a valid format or is corrupt. Recompile and relink.

CVPACK Fatal Error CK1007

invalid table *table* in module *module*

The given table in the given object file is not valid, or the module is corrupt. Recompile and relink. If the error recurs, try using different options.

CVPACK Fatal Error CK1008

cannot write output file

The disk is probably full. This error leaves a corrupt file on disk.

CVPACK Fatal Error CK1011

type information corrupt, recompile module *modulename*

The given object file contains corrupted information. Recompile. Try compiling with a different /debug keyword.

CVPACK Fatal Error CK1012

no debugging information

The executable file does not contain debugging information. Do one of the following:

- To put at least minimal debugging information into the executable file, link using the /DEBUG option.
- To include full debugging information in an object file, compile or assemble using the /Zi or /Z7 option.
- To include minimal information and line numbers in an object file, compile or assemble using the /Zd option.

CVPACK Fatal Error CK1013

incompatible version of debugging information

One of the following has occurred:

- The executable file was linked using an obsolete or unsupported linker.
- The executable file was already packed using a previous version of CVPACK.
- The executable file is corrupt. Recompile and relink.

CVPACK Fatal Error CK1017

cannot find precompiled types file; relink with filename

The program used a precompiled header, but the program was linked without filename. This file is the object file that was created when the header was precompiled. If the program uses a library that was created using precompiled headers, then filename was not linked from the library. One of the following

may be a solution:

- Extract the object from the library and link it to the program explicitly.
- Force LINK to include the object in the program by specifying the /INCLUDE option with a symbol from the object.
- Change the source code so that code or data in filename is referenced either from an object file that uses the library or from another module that is being linked from the library.

CVPACK Fatal Error CK1018

userfile cannot use precompiled types file *pctfile* : type counts do not match

The *userfile* uses the precompiled types file *pctfile*, but *pctfile* does not contain the correct number of types for *userfile*. Recompile *userfile* and relink. If a makefile is used, check the makefile dependencies.

CVPACK Fatal Error CK1021

userfile cannot use precompiled types file *pctfile* : signatures do not match

The *userfile* uses the precompiled types file *pctfile*, but the internal signature in *pctfile* does not match the internal signature in *userfile*. Probably *pctfile* was rebuilt but *userfile* was not rebuilt. Recompile *userfile* and relink. If a makefile is used, check the makefile dependencies.

CVPACK Fatal Error CK1022

cannot open program database *pdbfile*

CVPACK could not open the given program database (.PDB file). One of the following may be a cause:

- The .PDB file does not exist. Check the spelling of the filename and path.
- The .PDB file was opened or deleted by another process.

CVPACK and LINK look for .PDB files in the following locations in this order:

1. Where the .EXE or .DLL resides (the absolute path written in the .OBJ file).
2. Where the .OBJ or .DLL is opened from.

You cannot specify a .PDB file's name or location on the command line.

CVPACK Fatal Error CK1023

modulename cannot use program database *pdbfile* : module newer than database

The .OBJ file *modulename* uses the .PDB file *pdbfile*, but *pdbfile* is older than *modulename*. Recompile and relink. If a makefile is used, check the makefile dependencies.

CVPACK Fatal Error CK1024

modulename cannot use program database *pdbfile*: signatures do not match

The .OBJ file *modulename* uses the .PDB file *pdbfile*, but the internal signature

in *pdbfile* does not match the internal signature in *modulename*. Delete *modulename*, recompile, and relink. If a makefile is used, check the makefile dependencies.

CVPACK Fatal Error CK1025

incorrect COFF symbol table

CVPACK was unable to interpret the COFF symbols emitted by the compiler. Recompile and relink.

CVPACK Fatal Error CK1027

you have accessed an incorrect version of MSPDB60.DLL, recheck your installation

CVPACK Fatal Error CK1028

file compiled /Yc /Yu /Z7; cannot create debug info, recompile all with /Zi

You created two PCH files, one based on the other using the /Z7 compiler option. Only one precompiled type object can be referenced. The only way to use PCH files based on other PCH files is to use the /Zi compiler option to put all types in the .PDB file.

CVPACK Warning CK4001

file already packed

CVPACK took no action because the executable file has already been processed by this version of CVPACK.

CVPACK Warning CK4003

error in lexical scopes for module *module*; symbols deleted

The scoping of symbols in the given object *module* was corrupted. CVPACK deleted the symbols in the module. This is probably a compiler error. Recompile and relink the object file.

CVPACK Warning CK4004

unrecognized symbols discarded; recompile module *modulename*

The compiler generated an invalid symbol record in the *modulename* object file. CVPACK discarded the record. The debugger can run the resulting executable file. However, the symbol will be accessible only by the decorated name known to the linker, and display of the symbol will be limited. To see decorated names of identifiers, use DUMPBIN /UNDECORATE.

CVPACK Warning CK4005

unrecognized type converted to NULL; recompile module *modulename*

The compiler generated an invalid type record in the *modulename* object file. CVPACK discarded the record. The debugger can run the resulting executable file, but display of objects that reference the invalid type will be limited.

CVPACK Warning CK4006

type length limit exceeded; converted to NULL

A struct, enum, union, or class was too large for CVPACK to process. You can use the executable file with the debugger, but you cannot reference the object or type while debugging.

CVPACK Warning CK4007

unrecognized option /option; option ignored

The given option is not a valid CVPACK option. CVPACK ignored the specification and attempted to run using the rest of the command line.

CVPACK Warning CK4008

source line limit exceeded in module filename; source line info discarded

CVPACK could not translate the debugging information in the given filename module because filename was too large. CVPACK discarded the debugging information for this module. Break up the module into smaller files.

CVPACK Warning CK4009

exceeded 64K types at module; discarding subsequent types

The debugging information exceeded a CVPACK limit. One of the following may be a solution:

- Compile some object files without debugging information.
- Avoid including unneeded header files.

The checksum can be calculated only when running on Windows NT (including Windows 2000).

CVPACK Warning CK4010

cannot update checksum

The checksum can be calculated only when running on Windows NT (including Windows 2000).

CVTRES Messages CVT1100 through CVT4001

The following descriptions cover CVT1100 through warning CVT4001:

CVTRES Fatal Error CVT1100

duplicate resource -- type: *type*, name: *name*, language: *language*, flags: *flags*, size: *size*

The given resource was specified more than once.

CVTRES Fatal Error CVT1101

cannot open *filename* for reading

CVTRES could not open and read the given file. One of the following may be a cause:

- The file does not exist. Check the spelling of the filename and path.
- The file does not have read permission.
- The file was opened or deleted by another process.
- There was not enough space on disk.

CVTRES Fatal Error CVT1102

out of memory; *size* bytes required

There was not enough memory for CVTRES to complete the operation.

CVTRES Fatal Error CVT1103

cannot read *filename*

An unrecoverable error occurred when CVTRES attempted to read the given file. Possibly the file is corrupt.

CVTRES Fatal Error CVT1104

cannot get location in file

CVTRES could not determine the current location in the file. Possibly the file is corrupt.

CVTRES Fatal Error CVT1105

cannot seek in file

CVTRES could not go to a location in the file. Possibly the file is corrupt.

CVTRES Fatal Error CVT1106

cannot write to file

An unrecoverable error occurred when CVTRES attempted to write to the given file.

CVTRES Fatal Error CVT1107

filename is corrupt

The given file is not a valid resource file.

CVTRES Fatal Error CVT1108

cannot open filename for writing

CVTRES could not open and write to the given file. One of the following may be a cause:

- The file does not exist. Check the spelling of the filename and path.
- The file does not have write permission.
- The file was opened or deleted by another process.
- There was not enough space on disk.

CVTRES Warning CVT4001

machine type not specified; assumed *type*

CVTRES did not find a machine specification. It assumed the given machine *type*. If the default is incorrect, rerun CVTRES using the /MACHINE option.

Math Errors M6101 through M6205 (x86 only)

The following descriptions cover Math Errors M6101 through M6205, which can occur on x86 systems. All of these math errors terminate the program with an exit code of 3 (a severe error):

Math Error M6101

invalid

An invalid operation occurred. This error usually occurs when the operand is NaN (not a number) or infinity.

Math Error M6102

denormal

A very small floating-point number was generated, which may no longer be valid because of a loss of significance. Denormal floating-point exceptions are usually masked, causing them to be trapped and operated upon.

Math Error M6103

divide by 0

A floating-point operation attempted to divide by zero.

Math Error M6104

overflow

An overflow occurred in a floating-point operation.

Math Error M6105

underflow

An underflow occurred in a floating-point operation. Underflow floating-point exceptions are usually masked, causing the underflowing value to be replaced by 0.0.

Math Error M6106

inexact

Loss of precision occurred in a floating-point operation. This exception is usually masked. Many floating-point operations cause a loss of precision.

Math Error M6107

unemulated

An attempt was made to execute a coprocessor instruction that is invalid or is not supported by the emulator.

Math Error M6108

square root

The operand in a square-root operation was negative.

Note: The sqrt function in the C run-time library and the FORTRAN intrinsic

function SQRT do not generate this error. The C sqrt function checks the argument before performing the operation and returns an error value if the operand is negative. The FORTRAN SQRT function generates the Math Error M6201.

Math Error M6110

stack overflow

A floating-point expression caused a stack overflow on the 8087/287/387 coprocessor or the emulator. Stack-overflow floating-point exceptions are trapped up to a limit of seven levels in addition to the eight levels usually supported by the 8087/287/387 coprocessor.

Math Error M6111

stack underflow

A floating-point operation resulted in a stack underflow on the 8087/287/387 coprocessor or the emulator. This error is often caused by a call to a long double function that does not return a value.

Math Error M6201

'function' : `_DOMAIN` error

An argument to the given function was outside the domain of legal input values for that function. For example, the following statements generate this error:

```
result = sqrt(-1.0) // C statement  
result = SQRT(-1.0) ! FORTRAN statement
```

Math Error M6202

'function' : `_SING` error

An argument to the given function was a singularity value for this function. The function is not defined for that argument. For example, in FORTRAN the following statement generates this error:

```
result = LOG10(0.0)
```

Math Error M6203

'function' : `_OVERFLOW` error

The given function result was too large to be represented.

Math Error M6205

'function' : `_TLOSS` error

A total loss of significance (precision) occurred. This error may be caused by giving a very large number as the operand of sin, cos, or tan because the operand must be reduced to a number between 0 and 2π .

NMAKE Messages

The following topics cover NMAKE Error U1000 through Warning U4011:

- [NMAKE Errors U1000 to U1049](#)
- [NMAKE Errors U1050 to U1100](#)
- [NMAKE Error U2001 to Warning U4011](#)

NMAKE Error U1000 through U1050

The following topics cover NMAKE Error U1000 through U1049:

NMAKE Fatal Error U1000

syntax error : ')' missing in macro invocation

A left parenthesis, (, appeared without a matching right parenthesis,), in a macro invocation. The correct form is \$(name); \$n is allowed for one-character names.

NMAKE Fatal Error U1001

syntax error : illegal character '*character*' in macro

The given character appeared in a macro but was not a letter, number, or underscore (_). If the colon (:) is omitted in a macro expansion, the following error OCCURS: syntax error : illegal character '=' in macro

NMAKE Fatal Error U1002

syntax error : invalid macro invocation '\$'

A single dollar sign (\$) appeared without a macro name associated with it. The correct form is \$(name). To specify a dollar sign, use a double dollar sign (\$\$) or precede the dollar sign with a caret (^).

NMAKE Fatal Error U1003

syntax error : '=' missing in macro

A macro invocation contained a colon (:), which begins a substitution, but it did not contain an equal sign (=). The correct form is:

```
$(macroname:oldstring=newstring)
```

NMAKE Fatal Error U1004

syntax error : macro name missing

One of the following occurred:

- The name of a macro being defined was itself a macro invocation that expanded to nothing. For example, if the macro named ONE is undefined or has a null value, the following macro definition causes this error:

```
$(ONE)=TWO
```

- A macro invocation did not specify a name in the parentheses. The

following specification causes this error:

```
$( )
```

The syntax for using a macro is:

```
$(name)
```

NMAKE Fatal Error U1005

syntax error : text must follow ':' in macro

A string substitution was specified for a macro, but the string to be changed in the macro was not specified.

NMAKE Fatal Error U1006

syntax error : missing closing double quotation mark

An opening double quotation mark (") appeared without a closing double quotation mark.

NMAKE Fatal Error U1007

double quotation mark not allowed in name

The specified target name or filename contained a double quotation mark ("). Double quotation marks can surround a filename but cannot be contained within it.

NMAKE Fatal Error U1017

unknown directive '!directive'

The specified directive is not one of the recognized directives.

NMAKE Fatal Error U1018

directive and/or expression part missing

The directive was incompletely specified. The expression part of the directive is required.

NMAKE Fatal Error U1019

too many nested !IF blocks

The limit on nesting of !IF directives was exceeded. The !IF preprocessing directives include !IF, !IFDEF, !IFNDEF, !ELSE IF, !ELSE IFDEF, and !ELSE IFNDEF.

NMAKE Fatal Error U1020

end-of-file found before next directive

An expected directive was missing. For example, an !IF was not followed by an !ENDIF.

NMAKE Fatal Error U1021

syntax error : !ELSE unexpected

An !ELSE directive was found that was not preceded by an !IF directive, or the directive was placed in a syntactically incorrect place. The !IF preprocessing directives include !IF, !IFDEF, !IFNDEF, !ELSE IF, !ELSE IFDEF, and !ELSE IFNDEF.

NMAKE Fatal Error U1022

missing terminating character for string/program invocation : 'char'

One of the following occurred:

- The closing double quotation mark (") in a string comparison in a directive was missing.
- The closing bracket (]) in a program invocation in a directive was missing.

NMAKE Fatal Error U1023

syntax error in expression

An expression was invalid. Check the allowed operators and operator precedence.

NMAKE Fatal Error U1024

illegal argument to !CMDSWITCHES

An unrecognized command switch was specified.

NMAKE Fatal Error U1025

syntax error : !ENDIF unexpected

An !ENDIF directive was found that was not preceded by an !IF directive, or the directive was placed in a syntactically incorrect place. The !IF preprocessing directives include !IF, !IFDEF, !IFNDEF, !ELSE IF, !ELSE IFDEF, and !ELSE IFNDEF.

NMAKE Fatal Error U1031

filename missing (or macro is null)

An !INCLUDE directive was found, but the name of the file to be included was missing or a macro representing the filename expanded to nothing.

NMAKE Fatal Error U1033

syntax error : 'string' unexpected

The given string is not part of the valid syntax for a makefile. The following are examples of causes and results of this error:

- If the closing set of angle brackets (<<) for an inline file are not at the beginning of a line, the following error occurs: `syntax error : 'EOF' unexpected`
- If a macro definition in the makefile contained an equal sign (=) without a preceding name or if the name being defined is a macro that expands to nothing, the following error occurs: `syntax error : '=' unexpected`
- If the semicolon (;) in a comment line in TOOLS.INI is not at the beginning of the line, the following error occurs: `syntax error : ';' unexpected`
- If the makefile has been formatted by a word processor, the following error can occur: `syntax error : ':' unexpected`

NMAKE Fatal Error U1034

syntax error : separator missing

The colon (:) that separates targets and dependents is missing.

NMAKE Fatal Error U1035

syntax error : expected ':' or '=' separator

Either a colon (:) or an equal sign (=) was expected. Possible causes include the following:

- A target was not followed by a colon.
- A single-letter target was followed by a colon and no space (such as a:). NMAKE interpreted it as a drive specification.
- An inference rule was not followed by a colon.
- A macro definition was not followed by an equal sign.
- A character followed a backslash (\) that was used to continue a command to a new line.
- A string appeared that did not follow any NMAKE syntax rule.
- The makefile was formatted by a word processor.

NMAKE Fatal Error U1036

syntax error : too many names to left of '='

Only one string is allowed to the left of a macro definition.

NMAKE Fatal Error U1037

syntax error : target name missing

A colon (:) was found before a target name was found. At least one target is required.

NMAKE Fatal Error U1038

internal error : lexer

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1039

internal error : parser

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1040

internal error : macro expansion

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1041

internal error : target building

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1042

internal error : expression stack overflow

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1043

internal error : temp file limit exceeded

Note the circumstances of the error, try to isolate the problem, create a reproducible test case, and report to technical support.

NMAKE Fatal Error U1045

spawn failed : message

A program or command called by NMAKE failed for the given reason.

NMAKE Fatal Error U1047

argument before ')' expands to nothing

The parentheses following the preprocessing operator DEFINED or EXIST either were empty or contained an argument that evaluated to a null string.

NMAKE Fatal Error U1048

cannot write to file '*filename*'

NMAKE could not write to the given file. One cause of this error is a read-only file specified with /X.

NMAKE Fatal Error U1049

macro or inline file too long (maximum : 64K)

An inline file or a macro exceeded the limit of 64K.

[Return to Main NMAKE Messages Page](#)

NMAKE Error U1050 through U1100

The following topics cover NMAKE Error U1050 through U1100:

NMAKE Fatal Error U1050

message

The message specified with the !ERROR directive was displayed.

NMAKE Fatal Error U1051

out of memory

NMAKE ran out of memory, including virtual memory, because the makefile was too large or complex. One of the following may be a solution:

- Free some space on disk.
- Increase the size of the Windows NT (including Windows 2000) paging file,

or the Windows swap file.

- If only part of the makefile is being used, either divide the makefile into separate files or use !IF preprocessing directives to limit the amount that NMAKE must process. The !IF directives include !IF, !IFDEF, !IFNDEF, !ELSE IF, !ELSE IFDEF, and !ELSE IFNDEF.

NMAKE Fatal Error U1052

file '*filename*' not found

NMAKE could not find the given file, which was specified with one of the following:

- /F option
- !INCLUDE preprocessing directive
- At sign (@) specifier for a response file

Check that the file exists and the filename is spelled correctly.

NMAKE Fatal Error U1053

file '*filename*' unreadable

The file cannot be read. One of the following may be a cause:

- The file is in use by another process.
- A bad area exists on disk.
- A bad file-allocation table exists.

NMAKE Fatal Error U1054

cannot create inline file '*filename*'

NMAKE failed to create the given inline file. One of the following may be a cause:

- A file by that name exists with a read-only attribute.
- The disk is full.

NMAKE Fatal Error U1055

out of environment space

The operating system ran out of room for environment variables. Either increase the environment space or set fewer environment variables.

NMAKE Fatal Error U1056

cannot find command processor

The command processor was not in the path specified in the COMSPEC or PATH environment variables. NMAKE uses COMMAND.COM or CMD.EXE as a command processor when executing commands. It looks for the command processor first in the path set in COMSPEC. If COMSPEC does not exist, NMAKE searches the directories specified in PATH.

NMAKE Fatal Error U1057

cannot delete temporary file '*filename*'

NMAKE failed to delete the temporary inline file.

NMAKE Fatal Error U1058

terminated by user

NMAKE was halted by Ctrl+C or Ctrl+BREAK.

NMAKE Fatal Error U1059

syntax error : '}' missing in dependent

A search path for a dependent was incorrectly specified. Either a space existed in the path or the closing brace (}) was omitted. The syntax for a directory specification for a dependent is {directories}dependent where directories specifies one or more paths, each separated by a semicolon (;). No spaces are allowed. If part or all of a search path is replaced by a macro, make sure no spaces exist in the macro expansion.

NMAKE Fatal Error U1060

unable to close file : '*filename*'

NMAKE encountered an error while closing a file. One of the following may be a cause:

- The file is a read-only file.
- There is a locking or sharing violation.
- The disk is full.

NMAKE Fatal Error U1061

/F option requires a filename

The /F command-line option must be followed by either a makefile name or a dash (-), which represents standard input.

NMAKE Fatal Error U1062

missing filename with /X option

The /X command-line option requires the name of the file to which diagnostic error output should be redirected. To use standard output, specify a dash (-) as the output filename.

NMAKE Fatal Error U1063

missing macro name before '='

A macro definition on the NMAKE command line contained an equal sign (=) without a preceding name. This error can occur if the macro name being defined is itself a macro that expands to nothing.

NMAKE Fatal Error U1064

MAKEFILE not found and no target specified

The NMAKE command line did not specify a makefile or a target, and the current

directory did not contain a file named MAKEFILE. NMAKE requires either a makefile or a command-line target (or both). To make a makefile available to NMAKE, either specify the /F option or place a file named MAKEFILE in the current directory. NMAKE can create a command-line target by using an inference rule if a makefile is not provided.

NMAKE Fatal Error U1065

invalid option 'option'

The specified option is not a valid option for NMAKE.

NMAKE Fatal Error U1069

no match found for wildcard 'filename'

There is no file that matches the given filename, which was specified using one or more wildcards (* and ?). A target file that is specified using a wildcard must exist on disk.

NMAKE Fatal Error U1070

cycle in macro definition 'macroname'

The given macro definition contained a macro whose definition contained the given macro. Circular macro definitions are invalid. For example, the following macro definitions:

```
ONE=$( TWO )
TWO=$( ONE )
```

causes the following error:

```
cycle in macro definition 'TWO'
```

NMAKE Fatal Error U1071

cycle in dependency tree for target 'targetname'

A circular dependency exists in the dependency tree for the given target. The given target is a dependent of one of the dependents of the given target. Circular dependencies are invalid.

NMAKE Fatal Error U1072

cycle in include files : 'filename'

The given file includes a file that eventually includes the given file. Inclusions (using the !INCLUDE preprocessing directive) cannot be circular.

NMAKE Fatal Error U1073

don't know how to make 'targetname'

The specified target does not exist, and there is no command to execute or inference rule to apply. One of the following may be a solution:

- Check the spelling of the target name.
- If targetname is a pseudotarget, specify it as a target in another description block.

- If targetname is a macro invocation, be sure it does not expand to a null string.

NMAKE Fatal Error U1076

name too long

A string exceeded one of the following limits:

- A macro name cannot exceed 1024 characters.
- A target name (including its path) cannot exceed 256 characters.
- A command cannot exceed 2048 characters.

NMAKE Fatal Error U1077

'program' : return code *'value'*

The given command or program called by NMAKE failed and returned the given exit code. To suppress this error and continue the NMAKE session, use the /I option, the .IGNORE dot directive, or the dash (-) command modifier. To continue the NMAKE session for unrelated parts of the dependency tree, use the /K option.

NMAKE Fatal Error U1078

constant overflow at *'expression'*

The given expression contained a constant that exceeded the range - 2,147,483,648 to 2,147,483,647. The constant appeared in one of the following situations:

- An expression specified with a preprocessing directive
- An error level specified with the dash (-) command modifier

NMAKE Fatal Error U1079

illegal expression : divide by zero

An expression tried to divide by zero.

NMAKE Fatal Error U1080

operator and/or operand usage illegal

The expression incorrectly used an operator or operand. Check the allowed set of operators and their order of precedence.

NMAKE Fatal Error U1081

'filename' : program not found

NMAKE could not find the given program in order to run it. Make sure that the program is in a directory specified in the PATH environment variable and is not misspelled.

NMAKE Fatal Error U1082

'command' : cannot execute command; out of memory

There is not enough memory to execute the given *command*. One solution is to use the /N option to generate a batch file, then run the batch file instead of

using NMAKE. In most cases the results will be the same.

NMAKE Fatal Error U1083

target macro 'target' expands to nothing

The given target is an invocation of a macro that has not been defined or has a null value. NMAKE cannot process a null target.

NMAKE Fatal Error U1084

cannot create temporary file '*filename*'

NMAKE was unable to create the temporary file it needs when it processes the makefile. One of the following may be a cause:

- The file already exists with a read-only attribute.
- There is insufficient disk space to create the file.
- The directory specified in the TMP environment variable does not exist.

NMAKE Fatal Error U1085

cannot mix implicit and explicit rules

A target and a pair of inference-rule extensions were specified on the same line. Targets cannot be named in inference rules.

NMAKE Fatal Error U1086

inference rule cannot have dependents

The colon (:) in an inference rule must be followed by one of the following:

- Newline character
- Semicolon (;), which can be followed by a command
- Number sign (#), which can be followed by a comment

NMAKE Fatal Error U1087

cannot have : and :: dependents for same target

A target cannot be specified in both a single-colon (:) and a double-colon (::) dependency. To specify a target in multiple description blocks, use :: in each dependency line.

NMAKE Fatal Error U1088

invalid separator '::' on inference rule

An inference rule must be followed by a single colon (:).

NMAKE Fatal Error U1089

cannot have build commands for directive '*targetname*'

Dot directives cannot be followed by commands. The dot directives are .IGNORE, .PRECIOUS, .SILENT, and .SUFFIXES.

NMAKE Fatal Error U1090

cannot have dependents for directive '*targetname*'

Dot directives cannot be followed by dependents. The dot directives

are .IGNORE, .PRECIOUS, .SILENT, and .SUFFIXES.

NMAKE Fatal Error U1092

too many names in rule

An inference rule cannot specify more than two extensions.

NMAKE Fatal Error U1093

cannot mix dot directives

Multiple dot directives cannot be specified on one line. The dot directives are .IGNORE, .PRECIOUS, .SILENT, and .SUFFIXES.

NMAKE Fatal Error U1094

syntax error : only (NO)KEEP allowed here

Something other than KEEP or NOKEEP appeared after the closing set of angle brackets (<<) specifying an inline file. Only KEEP, NOKEEP, or a newline character may follow the angle brackets. No spaces, tabs, or other characters may appear. KEEP preserves the inline file on disk. NOKEEP deletes the file after the NMAKE session. The default is NOKEEP.

NMAKE Fatal Error U1095

expanded command line '*commandline*' too long

After macro expansion, the given command line exceeded the limit on length of command lines for the operating system. If the command is for a program that can accept command-line input from a file, change the command and supply input from either a file on disk or an inline file. For example, LINK and LIB accept input from a response file.

NMAKE Fatal Error U1096

cannot open inline file '*filename*'

NMAKE could not create the given inline file. One of the following occurred:

- The disk was full.
- A file with that name exists as a read-only file.

NMAKE Fatal Error U1097

filename-parts syntax requires dependent

The current dependency does not have either an explicit dependent or an implicit dependent. Filename-parts syntax, which uses the percent (%) specifier, represents components of the first dependent of the current target.

NMAKE Fatal Error U1098

illegal filename-parts syntax in '*string*'

The given string does not contain valid filename-parts syntax.

NMAKE Fatal Error U1099

stack overflow

The makefile being processed was too complex for the current stack allocation in

NMAKE. NMAKE has an allocation of 0x3000 (12K). To increase NMAKE's stack allocation, run the EDITBIN utility with a larger stack option:

`EDITBIN /STACK:stacksize NMAKE.EXE` where *stacksize* is a number greater than the current stack allocation in NMAKE.

NMAKE Fatal Error U1100

macro '*macroname*' is illegal in the context of batch rule '*rule*'

NMAKE generates this error when the command block of a batch-mode rule directly or indirectly references a special file macro that is not \$<. \$< is the only allowed macro for batch-mode rules.

[Return to Main NMAKE Messages Page](#)

NMAKE Error U2001 through Warning U4011

The following topics cover NMAKE Error U2001 through Warning U4011:

NMAKE Error U2001

no more file handles (too many files open)

NMAKE could not find a free file handle. One of the following may be a solution:

- Reduce recursion in the build procedures.
- In Windows 95 or Windows 98, increase the number of file handles by changing the FILES setting in CONFIG.SYS to allow a larger number of open files. FILES=50 is the recommended setting.

NMAKE Warning U4001

command file can be invoked only from command line

A command file, which is invoked by the at-sign (@) specifier, cannot contain a specification for another command file. Such nesting is not allowed. The specification was ignored.

NMAKE Warning U4002

resetting value of special macro '*macroname*'

The given predefined macro was redefined.

NMAKE Warning U4004

too many rules for target '*targetname*'

More than one description block was specified for the given target using single colons (:) as separators. NMAKE executed the commands in the first description block and ignored later blocks. To specify the same target in multiple dependencies, use double colons (::) as the separator in each dependency line.

NMAKE Warning U4005

ignoring rule '*rule*' (extension not in .SUFFIXES)

The given *rule* contained a suffix that is not specified in the .SUFFIXES list.

NMAKE ignored the *rule*. This warning appears only when the */P* option is used.

NMAKE Warning U4006

special macro undefined : '*macroname*'

The given special macro name is undefined and expands to nothing.

NMAKE Warning U4007

filename '*filename*' too long; truncating to 8.3

The base name of the given file has more than eight characters, or the extension has more than three characters. NMAKE truncated the name to an eight-character base and a three-character extension. If long filenames are supported by your file system, enclose the name in double quotation marks (").

NMAKE Warning U4008

removed target '*target*'

NMAKE was interrupted while trying to build the given target, and the target file was incomplete. Because the target was not specified in the .PRECIOUS list, NMAKE deleted the file.

NMAKE Warning U4010

'*target*' : build failed; */K* specified, continuing ...

A command in the commands block for the given *target* returned a nonzero exit code. The */K* option told NMAKE to continue processing unrelated parts of the build and to issue an exit code 1 when the NMAKE session is finished. If the given target is itself a dependent for another target, NMAKE issues warning U4011 after this warning.

NMAKE Warning U4011

'*target*' : not all dependents available; target not built

A dependent of the given *target* either did not exist or was out-of-date, and a command for updating the dependent returned a nonzero exit code. The */K* option told NMAKE to continue processing unrelated parts of the build and to issue an exit code 1 when the NMAKE session is finished. This warning is preceded by warning U4010 for each dependent that failed to be created or updated.

[Return to Main NMAKE Messages Page](#)

Profiler Errors PRF1005 through PRF4642

The following descriptions cover Profiler Errors PRF1005 through PRF4642:

Profiler Error PRF1005

fatal error -- common -- operating system ran out of memory

There is not enough memory to run this profiler module.

Profiler Error PRF1011

fatal error -- common -- cannot open file filename

The profiler module could not find the specified file.

Profiler Error PRF1012

fatal error -- common -- cannot read expected number of bytes from file filename

The profiler module could not read the expected number of bytes. This error is often caused by trying to profile an .EXE or .DLL file that was not linked with profiling enabled (/PROFILE linker command-line option).

Profiler Error PRF1013

fatal error -- common -- cannot write expected number of bytes to file filename

The profiler module could not write the expected number of bytes. Perhaps the disk is full or the file is read-only.

Profiler Error PRF1101

fatal error -- common -- missing option specifier

A / or - character was not followed by a command-line option specifier sequence. Check the command line.

Profiler Error PRF1102

fatal error -- common -- option: invalid option

An invalid option was found on the command line. Check the command line.

Profiler Error PRF1103

fatal error -- common -- option option requires an additional parameter.

This option was not followed by the correct number of parameters. Check the command line.

Profiler Error PRF1104

fatal error -- common -- option option requires a numeric parameter

A numeric parameter was expected for this option but was not found. Check the command line.

Profiler Error PRF1105

fatal error -- common -- option option specified more than number times

An option was specified too many times. Check the command line.

Profiler Error PRF1201

fatal error -- common -- missing filename after @

The @ character is supposed to be followed by a response filename. Check the command line.

Profiler Error PRF1202

fatal error -- common -- response files nested too deeply

Response files can be nested eight deep. Perhaps a response file calls itself recursively.

Profiler Error PRF1301

fatal error -- common -- file filename has an incorrect (old) signature

A PBI, PBO, PBT, _XE, or _LL file does not contain the correct header bytes. Perhaps the file is corrupted or was created by an old version of the profiler.

Profiler Error PRF1302

fatal error -- common -- PBI file filename has an incorrect (old) version number

A PBI file does not contain the correct version number. Perhaps the file is corrupted or was created by an old version of the profiler.

Profiler Error PRF1303

fatal error -- common -- PBT file filename has an incorrect (old) version number

A PBT file does not contain the correct version number. Perhaps the file is corrupted or was created by an old version of the profiler.

Profiler Error PRF1304

fatal error -- common -- PBT or PBO file filename conflicts with current profiling state

There was an attempt to merge a PBT or a PBO file into another PBT file with a different profiling mode selected.

Profiler Error PRF1306

fatal error -- common -- PBT or PBO file filename is not derived from same PBI file

The PBT, PBO, and PBI files are not properly synchronized. Run PREP Phase I again.

Profiler Error PRF1307

fatal error -- common -- CPU type of module module is different from initial one
CPU types for different modules don't match. All modules must be compiled for the same CPU type.

Profiler Error PRF1401

fatal error -- common -- filename: not a portable executable file

Function profiling only: The specified executable file is not a standard executable file for Windows NT 4.0 or Windows 2000. Perhaps it is a 16-bit executable for

Windows 3.x.

Profiler Error PRF1421

fatal error -- PREP I -- no input specified

There were no input modules specified for PREP Phase I. Check the command line.

Profiler Error PRF1422

fatal error -- PREP I -- syntax error in /EXC or /INC specification: specification

The /EXC and /INC options require function names or filenames. This specification could not be decoded. Check the command line.

Profiler Error PRF1423

fatal error -- PREP I -- number of modules exceeds number

The maximum number of modules has been exceeded. Restructure the merge procedure to include more stages.

Profiler Error PRF1424

fatal error -- PREP I -- Debugging information in filename is not the right format for profiling, relink with /PROFILE

Could not use debuggine information stored in the Program Database. Debug information, to be used, must be stored in the executable in old-style format. Relink with the /PROFILE option to do this.

Profiler Error PRF1425

fatal error -- PREP I -- filename: invalid runtime relocation, try relink

The profiler found bad relocation information in the program to be profiled. This is usually caused by using an obsolete version of the linker. Relink your code with the new profiler and run PREP again.

Profiler Error PRF1426

A fixed image executable cannot be profiled, relink file without the /fixed switch.

This error can also be caused by omitting the /PROFILE option on the LINK command line.

Profiler Error PRF1427

Executable image too large to profile - try breaking it down into smaller DLLs.

The executable cannot be profile at its present size.

Profiler Error PRF1441

fatal error -- PROFILE -- filename: not a valid Win32s executable file

Win32s only. The Win32s extension could not load this file. Perhaps the file is corrupted, has not been prepared for profiling, or is not a graphical user interface application. If the file were a 16-bit executable, error message 1401 would be issued.

Profiler Error PRF1442

fatal error -- PROFILE -- timestamp of executable has changed since PREP was run

The PBI file was created from an older EXE file. This means that the processed executable file has changed since it was originally created by PREP Phase I. Run PREP Phase I again.

Profiler Error PRF1443

fatal error -- PROFILE -- cannot run program filename

The profiler could not load the EXE file. Check that the original EXE file does run under Windows NT 4.0 or Windows 2000.

Profiler Error PRF1444

this version of Windows does not support this manner of profiling

The function timing (/ft) option is not supported for this version of Windows.

Profiler Error PRF1445

Line count and line coverage are not supported on Win32S.

The Win32S version of the profiler does not support line count or line coverage profiling.

Profiler Error PRF1461

fatal error -- PREP II -- no PBT output file specified

Neither the /OT nor /M option was specified. Check the command line.

Profiler Error PRF1462

fatal error -- PREP II -- no PBT input file specified

Neither the /IT nor /M option was specified. Check the command line.

Profiler Error PRF1463

fatal error -- PREP II -- module: different clock/sampling frequency

The sampling frequency for this module is different from that of other modules that were profiled. Rerun PROFILE again for all modules, using the same sampling frequency.

Profiler Error PRF1465

fatal error -- PREP II -- CPU type in PBT or PBO file filename conflicts with current one

Occurs only during PREP merging. The CPU type does not match the CPU type used to generate the incoming PBT or PBO files. All profiler steps must be run on the same type of computer.

Profiler Error PRF1481

fatal error -- PLIST -- option inconsistent with sorting type

The options for PLIST depend on the kind of profiling being done. This error occurs, for example, when the Sort By Name option is specified for a line profiling run.

Profiler Error PRF1482

fatal error -- PREP or PROFILE -- abnormal termination in PREP or PROFILE detected, PLIST cannot continue

The files output by PREP or PROFILE were not valid because one of these utilities did not complete its task. This error is usually caused by an error during the execution of PREP or PROFILE, such as a missing .MAP file.

Profiler Error PRF1504

fatal error -- common -- illegal fixup address in mapfile filename

The mapfile lists all link-time fixups (including all the function call sites). This error indicates that one of the fixups was illegal, perhaps because the mapfile was not generated by the same link as the corresponding EXE (or DLL). This error is often preceded by the warning (PRF4528) about the EXE and mapfile timestamps being out of sync.

Profiler Error PRF1521

fatal error -- PREP I -- no functions were marked for profiling in module module

Function profiling only: at least one function must be included for profiling in each module. Check the /INC, /EXC, and /EXCALL options on the command line.

Profiler Error PRF1522

fatal error -- PREP I -- more than one function matched entry specification: *funcname1* and *funcname2*

Function profiling only: there were two or more functions with matching names. Perhaps these are C++ overloaded functions. Use decorated names to specify the functions.

Profiler Error PRF1523

fatal error -- PREP I -- filename is not profiled, and can not import modified version version

For function-level profiles, PREP uses modified EXEs and DLLs, with modified names with extensions _XE and _LL. If any modules listed in the import table of a profiled module refer to another profiled module, but the imported module is itself not being profiled, then the import table of that imported module cannot be changed, and this error occurs. The solution is to include the offending module in the profile.

Profiler Error PRF1524

fatal error -- PREP I -- Debugging info in filename is not the right format for profiling, relink with /PROFILE

You must link executable files with profiling enabled (the /PROFILE linker option) to profile them. Change your linker options, rebuild, and re-run PREP.

Profiler Error PRF1542

fatal error -- PROFILE -- cannot set or delete environment variable var

Function profiling only: The specified environment variable was unavailable.

Profiler Error PRF1543

fatal error -- PROFILE -- NtQueryIntervalProfile: NT internal error number occurred

A Windows NT 4.0 or Windows 2000 error was detected.

Profiler Error PRF1544

fatal error -- PROFILE -- NtSetIntervalProfile: NT internal error number occurred

A Windows NT 4.0 or Windows 2000 error was detected.

Profiler Error PRF1545

fatal error -- PROFILE -- function sampling is no longer supported

Function sampling is not supported in this version. Use function timing to get more accurate results.

Profiler Error PRF1621

fatal error -- PREP I -- no debugging information in file filename

Line profiling only: debugging information must be included in the executable file. Run the compiler and linker again with the debugging options set.

Profiler Error PRF1622

fatal error -- PREP I -- no source lines marked for profiling in module module

Line profiling only: at least one line must be included for profiling in each module. Check the /INC, /EXC, and /EXCALL options on the command line.

Profiler Error PRF1641

fatal error -- PROFILE -- user-defined breakpoint found at address

Line profiling only: No breakpoints may be set in the executable modules being profiled. These are not breakpoints set by the debugger but rather embedded INT 3 instructions. This error is only fatal if the INT 3 instruction is encountered during the execution of your program.

Profiler Error PRF1642

fatal error -- PROFILE -- unexpected single-step exception occurred at address

Line profiling only: There was a problem with the executable file. Try relinking and rerunning PREP Phase I.

Profiler Error PRF1643

fatal error -- PROFILE -- an access violation occurred at address

Line profiling only: There was a problem with the executable file. Try relinking and rerunning PREP Phase I. Check the EXE file with the debugger.

Profiler Error PRF2541

error -- PROFILE -- cannot start sampler

Perhaps the executable program is too large.

Profiler Error PRF2542

error -- PROFILE -- cannot stop sampler
Perhaps the executable program is too large.

Profiler Error PRF4101

warning -- common -- option option overrides an earlier option

Two or more mutually exclusive switches were specified, or the same switch was specified more than once. The last switch takes precedence.

Profiler Error PRF4106

warning -- common -- function/source/object/lib name was never matched

The program element was not found. Make sure the name was spelled or decorated correctly.

Profiler Error PRF4107

warning -- common -- /SF option not valid for line-level or sample profiles

The start function option should be used only with function timing, function counting, or function coverage profiling.

Profiler Error PRF4111

warning -- common -- no error file (/E) specified for /A option

The /A option cannot be used without the /E option. The /A option is ignored and the error output sent to stdout.

Profiler Error PRF4421

warning -- PREP I -- an offset duplication was detected

There appears to be a problem with the (codeview) line number debug information. Perhaps there is an aliasing problem.

Profiler Error PRF4422

warning -- PREP I -- filename imports unknown module module

PREP searches for imported modules (mostly DLLs) in the following sequence:

- Directory PREP itself lives in
- Current directory
- Windows directories
- Directories on the path

Because the current directory depends on the profiler rather than on the profiled program, all modules might not be found. In most cases, this warning is not serious. However, if profiling leads to a GP fault (or to incorrect behavior), it might be necessary to include the "unknown module" in the profile.

Profiler Error PRF4423

warning -- PREP I -- filename: ignoring unknown runtime relocation type, verify linker version

(MIPS Specific) The profiler found bad relocation information in the program to

be profiled. This is usually caused by using an obsolete version of the linker. Relink your code with the new profiler, and run PREP again.

Profiler Error PRF4481

warning -- PLIST -- clock frequency is zero

The clock frequency field in the PBT file has not been filled in yet, perhaps because the user has not done a profile yet, or has not merged his PBO file into his PBT file. Running PREP /M should fix the problem.

Profiler Error PRF4521

warning -- PREP I -- INIT environment variable not defined

The INIT environment variable should point to the directory that contains TOOLS.INI. Set the INIT environment variable to the directory containing your TOOLS.INI and rerun PREP.

Profiler Error PRF4522

warning -- PREP I -- 'publics' section not found in mapfile filename

The mapfile is corrupt.

Profiler Error PRF4523

warning -- PREP I -- 'static symbols' section not found in mapfile filename

The mapfile is corrupt.

Profiler Error PRF4524

warning -- PREP I -- entry function function not found

PREP could not find the specified start function (entry function) in the mapfile (see /SF switch). Perhaps the function name is misspelled.

Profiler Error PRF4525

warning -- PREP I -- cannot profile function function in file filename

A special function such as setjmp was specified.

Profiler Error PRF4526

warning -- PREP I -- module module already includes profiler DLL filename

The specified EXE or DLL already imports from PROFILER.DLL. The EXE (or DLL) may have been modified by PREP using the /OM switch and then renamed (from app._XE to app.EXE) before PREP was run again. Make sure that the module (EXE or DLL) specified is the original module and a module modified by PREP.

Profiler Error PRF4527

warning -- PREP I -- preferred load address not found in mapfile filename;
assumed address

Current versions of the linker list the preferred load address (that is, when the EXE gets loaded at this address none of the run-time relocation addresses need to be fixed up). PREP needs this information. Somehow this information was missing from the mapfile. If the latest linker was used, the mapfile was probably edited. Regenerate the mapfile, retaining the line that states the preferred load

address.

Profiler Error PRF4528

warning -- PREP I -- timestamps of mapfile filename and executable file do not match

The mapfile is not properly synchronized with the executable file. Relink and run PREP again.

Profiler Error PRF4541

warning -- PROFILE -- module not found in PBI file filename

An attempt was made to profile a program, but the EXE was not modified by PREP. This warning occurs while selectively profiling a DLL that is called by the EXE.

Profiler Error PRF4621

warning -- PREP I -- line number duplicated in file filename

There is a problem with the line number debug information. The same line number was identified with two different addresses. This is generally not serious.

Profiler Error PRF4641

warning -- PROFILE -- ReadProcessMemory: internal NT error number occurred

An internal error occurred. The profiler results may be meaningless.

Profiler Error PRF4642

warning -- PROFILE -- WriteProcessMemory: internal NT error number occurred

An internal error occurred. The profiler results may be meaningless.

Resource Compiler Messages

The following topics cover Resource Compiler Errors RC1000 through Warning RC4413:

- [Resource Compiler Errors RC1000 through RC1208](#)
- [Resource Compiler Errors RC2001 through RC2129](#)
- [Resource Compiler Errors RC2130 through RC2236](#)
- [Resource Compiler Warnings RC4000 through RC4413](#)

For information about Resource Compiler RWxxxx messages, see:

- [Resource Compiler Messages \(RWxxxx\)](#)

Resource Compiler Errors RC1000 through RC1208

The following topics cover Resource Compiler Errors RC1000 through RC1208:

Resource Compiler Fatal Error RC1000

UNKNOWN FATAL ERROR

Note the circumstances of the error, try to isolate the problem, and create a reproducible test case, then report this information to technical support.

Resource Compiler Fatal Error RC1002

out of heap space

The Resource Compiler ran out of storage for items that it stores in its heap. One of the following may be a solution:

- Increase the Windows swap file space. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory, reboot, and retry.
- Split the current file into two or more files and compile them separately.

Resource Compiler Fatal Error RC1003

error count exceeds number; stopping compilation

The number of errors was too great to continue compilation. Fix some errors and recompile.

Resource Compiler Fatal Error RC1004

unexpected end of file found

This error can be caused by omitting the linefeed and carriage return characters on the last line of a text file.

Resource Compiler Fatal Error RC1005

file not found: *filename*

The file specified in the RC command line was not found. Check to see whether the file has been moved to another directory and whether the filename or path is typed correctly. Files are searched for using the INCLUDE environment variable or the Directories setting.

Resource Compiler Fatal Error RC1007

unrecognized option '*option*' in '*filename*'

The specified option is not valid.

Resource Compiler Fatal Error RC1008

no input file specified

One or more source files must be specified on the Resource Compiler command line.

Resource Compiler Fatal Error RC1009

compiler limit : macros too deeply nested '*macro*'

The Resource Compiler limit for macro nesting was exceeded. Revise the specified source file to decrease the nesting depth of its macros.

Resource Compiler Fatal Error RC1010

no output file specified

The Resource Compiler command line did not specify a filename for the compiled resource file.

Resource Compiler Fatal Error RC1011

compiler limit : '*identifier*' : macro definition too big

The macro definition was longer than allowed. Try to split the definition into shorter definitions.

Resource Compiler Fatal Error RC1012

unmatched parenthesis : missing '*character*'

The parentheses in a preprocessor directive were not matched. The missing character is either a left or right parenthesis.

Resource Compiler Fatal Error RC1013

mismatched parentheses

Make sure that every open parenthesis has a matching closing parenthesis.

Resource Compiler Fatal Error RC1014

too many include files : depth = '*level*'

The nesting depth of #include directives was too great. Open files can be included by the use of nested directives. The source file containing the directive is counted as one of the files.

Resource Compiler Fatal Error RC1015

cannot open include file '*filename*'

The given include file either did not exist, could not be opened, or was not

found. Make sure that the environment settings are valid and that the correct path for the file is specified. Ensure that sufficient file handles are available to the Resource Compiler. If the file is on a network drive, make sure that you have permissions to open the file.

Resource Compiler Fatal Error RC1016

`#if[n]def` expected an identifier

The `#ifdef` or `#ifndef` conditional compilation directive was not supplied with an identifier to evaluate. An identifier must be specified with the `#ifdef` and `#ifndef` directives.

Resource Compiler Fatal Error RC1017

invalid integer constant expression

The expression in an `#if` directive either did not exist or did not evaluate to a constant.

Resource Compiler Fatal Error RC1018

unexpected `#elif`

The `#elif` directive did not appear within an `#if`, `#ifdef`, or `#ifndef` construct. Make sure that there is an `#if`, `#ifdef`, or `#ifndef` statement in effect before this statement.

Resource Compiler Fatal Error RC1019

unexpected `#else`

The `#else` directive did not appear within an `#if`, `#ifdef`, or `#ifndef` construct. Make sure that there is an `#if`, `#ifdef`, or `#ifndef` statement in effect before this statement.

Resource Compiler Fatal Error RC1020

unexpected `#endif`

An `#endif` directive appeared without a matching `#if`, `#ifdef`, or `#ifndef` directive. Make sure that there is a matching `#endif` for each `#if`, `#ifdef`, and `#ifndef` statement.

Resource Compiler Fatal Error RC1021

invalid preprocessor command `'string'`

The characters following the number sign (`#`) did not form a valid preprocessor directive. The number sign cannot be used as the first character in an identifier.

Resource Compiler Fatal Error RC1022

expected `#endif`

An `#if`, `#ifdef`, or `#ifndef` directive was not terminated with an `#endif` directive. Make sure that there is an `#if`, `#ifdef`, or `#ifndef` statement in effect before this statement.

Resource Compiler Fatal Error RC1023

cannot open source file `'filename'`

The specified file either did not exist, could not be opened, or was not found. Make sure that the environment settings are valid and that the correct path for the file is specified. If this error appears without an error message, the Resource Compiler has run out of file handles.

Resource Compiler Fatal Error RC1047

"too many option options, '*string*'"

The given option was specified too many times. The given string is the argument to the option that caused the error.

Resource Compiler Fatal Error RC1048

unknown option '*character*' in '*option*'

The given character was not a valid letter for the option.

Resource Compiler Fatal Error RC1049

invalid numerical argument '*string*'

The Resource Compiler expected a numerical argument but received the given string. This error may be caused by giving a hexadecimal number without the necessary `\x` prefix or by a misformed floating-point number.

Resource Compiler Fatal Error RC1052

compiler limit : `#if` or `#ifdef` blocks nested too deeply

The program exceeded the maximum allowable nesting levels for `#if` and `#ifdef` directives. This error can be caused by include files that use these preprocessor directives.

Resource Compiler Fatal Error RC1067

compiler limit : identifier overflowed internal buffer

An internal compiler limit was exceeded.

Resource Compiler Fatal Error RC1101

no resource binary filename specified

The Rename Output (`/fo`) option was specified, but was not followed by filename. The following syntax must be used with the `/fo` option:

RC `/fooutfilefilename infilename.rc`

Resource Compiler Fatal Error RC1102

internal error : too many arguments to RCPP

Too many arguments were passed to the Resource Compiler preprocessor. Reduce the number of symbols defined with the Define Symbols (`/d`) option by defining them in your source. This error can also be caused by specifying too many include file search paths using the Include Search Path option (`/i`).

Resource Compiler Fatal Error RC1103

invalid switch, option

The specified option is not a valid Resource Compiler option.

Resource Compiler Fatal Error RC1105

invalid switch, option: too many /d switches

Too many symbols were defined using the Define Symbols (/d) option. Define some symbols in your source and recompile.

Resource Compiler Fatal Error RC1106

invalid switch: option

The specified option is not a valid Resource Compiler option. The following options are valid:

/r Emit .RES file
/v Verbose (print progress messages)
/d Define a symbol
/fo Rename .RES file
/l Default language ID in hex
/i Add a path for INCLUDE searches
/x Ignore INCLUDE environment variable
/c Define a code page used by NLS Conversion

Resource Compiler Fatal Error RC1107

invalid usage; use RC /? for Help

An invalid Resource Compiler option was specified. Valid options are one of the following: /r, /v, /d, /fo, /l, /i, /x, or /c:

Resource Compiler Fatal Error RC1109

error creating resource-name

Could not create specified .RES file. Make sure it is not being created on a read-only drive. Use the /V option to find out whether the file is being created.

Resource Compiler Fatal Error RC1110

could not open filename

RC could not open the specified resource script file. Make sure that the file exists.

Resource Compiler Fatal Error RC1116

RC terminating after preprocessor errors

The Resource Compiler halted due to other errors. Fix the other errors and recompile.

Resource Compiler Fatal Error RC1120

out of memory, needed number bytes

The Resource Compiler ran out of storage for items that it stores in its heap. Usually this is the result of having too many symbols. One of the following may be a solution:

- Increase the Windows swap file space. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory,

- reboot, and retry.
- Eliminate unnecessary include files, especially unneeded #defines and function prototypes.
- Split the current file into two or more files and compile them separately.

Resource Compiler Fatal Error RC1121

I/O error reading file

The Resource Compiler was not able to read a file. Check that the drive containing the file is available and that the file is valid.

Resource Compiler Fatal Error RC1122

I/O error writing file

The Resource Compiler could not write to a file. This error may be caused by insufficient disk space. You must have free space equivalent to about twice the size of the executable file you are creating. Other possible reasons for this error are:

- Writing to a read-only volume
- Writing to a bad sector
- A sharing violation

Resource Compiler Fatal Error RC1123

I/O error seeking in file

The compiler was unable to complete an I/O operation.

Resource Compiler Fatal Error RC1201

invalid switch - missing include path after /i

The Specify Include Path (/i) option was specified, but was not followed by a path.

Resource Compiler Fatal Error RC1202

invalid switch - missing default language ID after /l

The Specify Default Language (/l) option was specified, but was not followed by a hexadecimal language ID.

Resource Compiler Fatal Error RC1203

invalid hexadecimal default language ID specified.

The Specify Default Language (/l) option was specified, but was followed by an invalid hexadecimal language ID. See the Platform SDK for a list of valid language IDs.

Resource Compiler Fatal Error RC1204

Invalid switch - missing code page after /c

The Specify Code Page (/c) option was specified, but it was not followed by a code page.

Resource Compiler Fatal Error RC1205

invalid code page

The Specify Code Page (/c) option was specified, but was followed by an invalid code page. See Code Pages in the the online documentation for more information.

Resource Compiler Fatal Error RC1206

specified code page at cmd line does not exist in registry

The Specify Code Page (/c) option was specified, but was followed by a code page whose location is not specified in the registry.

Resource Compiler Fatal Error RC1207

default code page is invalid

The code page specified by the RCCODEPAGE environment variable is not valid. See Code Pages in the online documentation for more information.

Resource Compiler Fatal Error RC1208

input file has .RES extension

The .RES extension is used for Resource Compiler output. The .RC extension should be used for Resource Compiler scripts.

[Return to Main Resource Compiler Messages Page](#)

Resource Compiler Errors RC2001 through RC2129

Resource Compiler Error RC2001

newline in constant

A string constant was continued on a second line without either a backslash (\) or closing and opening double quotation marks ("). To break a string constant that is on two lines in the source file, do one of the following:

- End the first line with the line-continuation character, a backslash.
- Close the string on the first line with a double quotation mark and open the string on the next line with another quotation mark.

It is not sufficient to end the first line with \n, the escape sequence for embedding a newline character in a string constant.

Resource Compiler Error RC2003

expected '*defined id*'

An identifier was expected after the specified preprocessing keyword.

Resource Compiler Error RC2004

expected '*defined(id)*'

An identifier was expected after the left parenthesis following the specified preprocessing keyword.

Resource Compiler Error RC2005

#line expected a line number, found '*token*'

A *#line* directive lacked the required line-number specification.

Resource Compiler Error RC2006

#include expected a filename, found '*token*'

An *#include* directive lacked the required filename specification.

Resource Compiler Error RC2007

#define syntax

An identifier was expected following *#define* in a preprocessing directive.

Resource Compiler Error RC2008

'*character*' : unexpected in macro definition

The given character was found immediately following the name of the macro.

Resource Compiler Error RC2009

reuse of macro formal '*identifier*'

The given identifier was used more than once in the formal parameter list of a macro definition.

Resource Compiler Error RC2010

'*character*' : unexpected in formal list

The given character was used incorrectly in the formal parameter list of a macro definition.

Resource Compiler Error RC2012

missing name following '<'

An *#include* directive lacked the required filename specification.

Resource Compiler Error RC2013

missing '>'

The closing angle bracket (>) was missing from an *#include* directive.

Resource Compiler Error RC2014

preprocessor command must start as first non-whitespace

Non-white-space characters appeared before the number sign (#) of a preprocessor directive on the same line.

Resource Compiler Error RC2015

too many chars in constant

A character constant contained more than two characters. Character constants are limited to one character (standard character constants) or two characters (long character constants). Note that an escape sequence (for example, \t for tab) is converted to a single character.

Resource Compiler Error RC2016

no closing single quote

A newline character was found before the closing single quotation mark of a character constant.

Resource Compiler Error RC2017

illegal escape sequence

An escape sequence appeared where one was not expected. An escape sequence - a backslash (\) followed by a number or letter - may occur only in a character or string constant.

Resource Compiler Error RC2018

unknown character '*hexnumber*'

The ASCII character corresponding to the given hexadecimal number appeared in the source file but is an illegal character. One possible cause of this error is corruption of the source file.

Resource Compiler Error RC2019

expected preprocessor directive, found '*character*'

The given character followed a number sign (#), but it was not the first letter of a preprocessor directive.

Resource Compiler Error RC2020

illegal digit number for base radix

The specified digit is not a valid digit for the base specified by radix. Either the digit or the radix could be incorrect. Octal digits must be numbers from 0 to 7, and hexadecimal digits must be numbers from 0 to 9 or letters from A through E.

Resource Compiler Error RC2021

expected exponent value, not '*character*'

The given character was used as the exponent of a floating-point constant but was not a valid number.

Resource Compiler Error RC2022

'number' : too big for character

The octal number following a backslash (\) in a character or string constant was too large to be represented as a character.

Resource Compiler Error RC2101

Invalid directive in preprocessed RC file

The RC file contains a #pragma directive. Use the #ifndef preprocessor directive with the RC_INVOKED constant that the Resource Compiler defines when it processes an include file. Place the #pragma directive inside a block of code that is not processed when the RC_INVOKED constant is defined. Code in the block is processed only by the C/C++ compiler and not by the Resource Compiler. The

following sample code demonstrates this technique:

```
#ifndef RC_INVOKED
#pragma pack(2) // C/C++ only, ignored by Resource Compiler
#endif
```

The `#pragma` preprocessor directive has no meaning in an `.RC` file. The `#include` preprocessor directive is used frequently in an `.RC` file to include a header file (either a project-based custom header file or a standard header file provided by Microsoft with one of its products). Some of these include files contain the `#pragma` directive. Because a header file can include one or more other header files, the file that contains the offending `#pragma` directive may not be immediately obvious. The `#ifndef RC_INVOKED` technique can control including header files in project-based header files.

Resource Compiler Error RC2102

string literal too long

An RCDATA string exceeded the maximum allowable length.

Resource Compiler Error RC2103

unexpected end of file in string literal

An end of file was found before the end of a string. The string is probably missing a closing double quotation mark (`"`).

Resource Compiler Error RC2104

undefined keyword or key name: key

The specified keyword or key name was not defined. If you encounter the following error message:

```
undefined keyword or key name: MFT_STRING
```

Open the file `...\mfc\include\AfxRes.h` and add the following include directive:

```
#include <winresrc.h>
```

Resource Compiler Error RC2105

BEGIN expected in string table

The `BEGIN` keyword must immediately follow the `ACCELERATORS` keyword.

Resource Compiler Error RC2106

BEGIN expected in accelerator table

The `BEGIN` keyword must immediately follow the `ACCELERATORS` keyword.

Resource Compiler Error RC2107

expected numeric command value

RC was expecting a numeric `idvalue` field in the `ACCELERATORS` statement. Make sure that you have used a `#define` constant to specify the value and that the constant is spelled correctly.

Resource Compiler Error RC2108

unbalanced parentheses

An open parenthesis was not matched with a closing parenthesis.

Resource Compiler Error RC2109

expected numerical dialog constant

A DIALOG statement requires integer values for the x, y, width, and height fields. Make sure these values are included after the DIALOG keyword and that they are not negative.

Resource Compiler Error RC2110

expected numerical dialog constant

A DIALOG statement requires integer values for the x, y, width, and height fields. Make sure these values are included after the DIALOG keyword and that they are not negative.

Resource Compiler Error RC2111

invalid control type

Each CONTROL statement in a DIALOG statement must be one of the following: 3STATE, AUTO3, AUTOCHECK, AUTORADIO, BEDIT, CHECKBOX, COMBOBOX, CONTROL, CTEXT, DEFPUSHBUTTON, EDITTEXT, GROUPBOX, HEDIT, ICON, IEDIT, LISTBOX, LTEXT, PUSHBOX, PUSHBUTTON, RADIOBUTTON, RTEXT, SCROLLBAR, USERBUTTON. Make sure these CONTROL statements are spelled correctly.

Resource Compiler Error RC2112

BEGIN expected in dialog

The BEGIN keyword must immediately follow the DIALOG keyword.

Resource Compiler Error RC2113

END expected in dialog

The END keyword must occur at the end of a DIALOG statement. Make sure there are no open quotes left from the preceding statement.

Resource Compiler Error RC2114

expected control class name

The class field of a CONTROL statement in the DIALOG statement must be one of the following types: BUTTON, COMBOBOX, EDIT, LISTBOX, SCROLLBAR, STATIC, or user-defined. Make sure the class is spelled correctly.

Resource Compiler Error RC2115

text string or ordinal expected in control

The text field of a CONTROL statement in the DIALOG statement must be either a text string or an ordinal reference to the type of control is expected. If using an ordinal, make sure that you have a #define statement for the control.

Resource Compiler Error RC2116

expecting number for ID

Expecting a number for the id field of a control statement in the DIALOG statement. Make sure you have a number or #define statement for the control ID.

Resource Compiler Error RC2117

expected numeric point size

The pointsize field of the FONT option in the DIALOG statement must be an integer point size value.

Resource Compiler Error RC2118

expected font face name

The typeface field of the FONT option in the DIALOG statement must be an ASCII character string enclosed in double quotation marks. This field specifies the name of a font.

Resource Compiler Error RC2119

expecting quoted string in dialog title

The captiontext field of the CAPTION option in the DIALOG statement must be an ASCII character string enclosed in double quotation marks.

Resource Compiler Error RC2120

expecting quoted string in dialog class

The class field of the CLASS option in the DIALOG statement must be an integer or a string, enclosed in double quotation marks.

Resource Compiler Error RC2121

BEGIN expected in menu

The BEGIN keyword must immediately follow the MENU keyword.

Resource Compiler Error RC2122

unknown menu subtype

The item-definition field of the MENU statement can contain only MENUITEM and POPUP statements.

Resource Compiler Error RC2123

END expected in menu

The END keyword must come at the end of a MENU statement. Make sure you do not have any open quotation marks or a mismatched pair of BEGIN and END statements.

Resource Compiler Error RC2124

empty menus not allowed

An END keyword appears before any menu items are defined in the MENU statement. Empty menus are not permitted by the Resource Compiler. Make

sure you do not have any open quotation marks within the MENU statement.

Resource Compiler Error RC2125

expected ID value for menuitem

The MENU statement must contain a menuID field, which specifies the name or number that identifies the menu resource.

Resource Compiler Error RC2126

expected menu string

Each MENUITEM and POPUP statement must contain a text field, which is a string enclosed in double quotation marks that specifies the name of the menu item or pop-up menu. A MENUITEM SEPARATOR statement requires no quoted string.

Resource Compiler Error RC2127

version WORDs separated by commas expected

Version numbers in a version resource should be of type WORD, and separated by commas.

Resource Compiler Error RC2128

DWORD expected

A valid version number was not found. It must be a DWORD type.

Resource Compiler Error RC2129

BEGIN expected in VERSIONINFO resource

The BEGIN keyword must immediately follow the VERSIONINFO keyword.

[Return to Main Resource Compiler Messages Page](#)

Resource Compiler Errors RC2130 through RC2236

Resource Compiler Error RC2130

#line expected a string containing the file name, found '*token*'

A #line directive was missing a required filename.

Resource Compiler Error RC2131

expecting quoted string for key

The field for this statement must be a character string enclosed in double quotation marks.

Resource Compiler Error RC2132

expected VALUE, BLOCK, or, END keyword

A block in a VERSION resource doesn't end properly, or a new block doesn't begin properly. This results in a VERSION statement that is not a valid block.

Resource Compiler Error RC2133

unexpected value in value data

The raw-data values in the RCDATA statement must be integers or strings, each separated by a comma. Make sure you did not leave out a comma or leave out a quotation mark around a string.

Resource Compiler Error RC2134**BEGIN expected in RCDATA**

The BEGIN keyword must immediately follow the RCDATA keyword.

Resource Compiler Error RC2135**file not found: filename**

The file specified in the RC command line was not found. Check to see whether the file has been moved to another directory and whether the filename or path is typed correctly. Files are searched for using the INCLUDE environment variable or the visual development environment INCLUDE directory setting (Tools menu, Options item, Directories tab).

Resource Compiler Error RC2136**missing '=' in EXSTYLE=<flags>**

An equal sign (=) was missing from an EXSTYLE (Extended Style Flags) statement. When the EXSTYLE is embedded in the DIALOG or MENU statement it must have the following form: EXSTYLE=FLAGS

Resource Compiler Error RC2137**empty character constant**

A pair of single quotes was found with no character specified between them.

Resource Compiler Error RC2138**unmatched close comment '/'***

An open comment sequence (/*) was not matched with a close comment sequence (*/) This error can be caused by nesting comments.

Resource Compiler Error RC2139**VERSION not a number**

A VERSION resource must be a number.

Resource Compiler Error RC2140**CHARACTERISTICS not a number**

A CHARACTERISTICS resource must be a number.

Resource Compiler Error RC2141**invalid type**

A different type was expected.

Resource Compiler Error RC2142**ellipsis requires three periods**

An incorrect number of periods (.) was used in an ellipsis.

Resource Compiler Error RC2143

font names must be ordinals

The pointsize field in the FONT statement must be an integer, not a string.

Resource Compiler Error RC2144

PRIMARY LANGUAGE ID not a number

The PRIMARY LANGUAGE ID must be a hexadecimal language ID. See Language and Country Strings in the online documentation for a list of valid Language IDs.

Resource Compiler Error RC2145

PRIMARY LANGUAGE ID too large

The PRIMARY LANGUAGE ID must be a valid hexadecimal language ID. See Language and Country Strings in the Ronline documentation for a list of valid Language IDs.

Resource Compiler Error RC2146

missing COMMA in LANGUAGE statement

The comma separating the primary language and secondary language is missing. The LANGUAGE statement must use the following syntax: LANGUAGE primary_language_ID,secondary_language_ID

Resource Compiler Error RC2147

SUBLANGUAGE ID not a number

The SUBLANGUAGE ID value must be a number. The LANGUAGE statement must use the following syntax: LANGUAGE primary_language_ID,secondary_language_ID Valid SUBLANGUAGE IDs are defined as SUBLANG_ constants in the WINNT.H file.

Resource Compiler Error RC2148

SUBLANGUAGE ID too large

The SUBLANGUAGE ID value was out of range. The LANGUAGE statement must use the following syntax: LANGUAGE primary_language_ID,secondary_language_ID Valid SUBLANGUAGE IDs are defined as SUBLANG_ constants in the WINNT.H file.

Resource Compiler Error RC2149

expected numeric constant in string table

A numeric constant, defined in a #define statement, must immediately follow the BEGIN keyword in a STRINGTABLE statement.

Resource Compiler Error RC2150

expected string in STRINGTABLE

A string is expected after each stringid value in a STRINGTABLE statement.

Resource Compiler Error RC2151

cannot re-use string constants

You are using the same value twice in a STRINGTABLE statement. Make sure you are not mixing overlapping decimal and hexadecimal values. Each ID in a STRINGTABLE must be unique. For maximum efficiency use contiguous constants that start on a multiple of 16.

Resource Compiler Error RC2152

invalid control character

A control character in the ACCELERATORS statement is invalid. A valid control character consists of one letter (only) following a caret (^).

Resource Compiler Error RC2153

hex constants must have at least 1 hex digit

An empty hexadecimal constant was found.

Resource Compiler Error RC2154

control character out of range [^A - ^Z]

A control character in the ACCELERATORS statement is invalid. The character following the caret (^) must be between A and Z, inclusive.

Resource Compiler Error RC2155

invalid accelerator

An event field in the ACCELERATORS statement was not recognized or was more than two characters in length.

Resource Compiler Error RC2156

expected string or constant accelerator command

RC was not able to determine what kind of key is being set up for the accelerator. The event field in the ACCELERATORS statement might be invalid.

Resource Compiler Error RC2157

expected comma in accelerator table

The Resource Compiler requires a comma between the event and idvalue fields in the ACCELERATORS statement.

Resource Compiler Error RC2159

invalid accelerator type [ASCII or VIRTKEY]

The type field in the ACCELERATORS statement must contain either the ASCII or VIRTKEY value.

Resource Compiler Error RC2160

cannot occur at the beginning of a macro definition

A macro definition began with a token-pasting operator (##).

Resource Compiler Error RC2161

cannot occur at the end of a macro definition

A macro definition ended with a token-pasting operator (##).

Resource Compiler Error RC2162

expected macro formal parameter

The token following a stringizing operator (#) was not a formal parameter name.

Resource Compiler Error RC2163

accelerator type required [ASCII or VIRTKEY]

The type field in the ACCELERATORS statement must contain either the ASCII or VIRTKEY value.

Resource Compiler Error RC2164

unexpected value in RCDATA

The raw-data values in the RCDATA statement must be integers or strings, each separated by a comma. Make sure you did not leave out a comma or leave out a quotation mark around a string.

Resource Compiler Error RC2165

string not found in DLGINCLUDE statement

The statement did not specify a valid include file. The DLGINCLUDE statement must use the following syntax: DLGINCLUDE "filename.h"

Resource Compiler Error RC2166

numeric value expected at line

The resource on the specified line must be a numeric value.

Resource Compiler Error RC2167

unrecognized VERSIONINFO field; BEGIN or comma expected

An unrecognized field was found in the FIXED part of a VERSIONINFO structure declaration. A VERSIONINFO field must be DWORDS separated by a comma.

Resource Compiler Error RC2168

resource too large

The size limitation of a resource was exceeded. This limit does not apply to cursors, icons, bitmaps, or other file-based resources.

Resource Compiler Error RC2169

resource file filename is not in 2.03 format

The specified resource used a format earlier than version 2.03. The resource file must be converted or re-created using the format for version 3.00 or later.

Resource Compiler Error RC2170

bitmap file filename is not in 3.00 format

Bitmaps using the Windows version 2.x format cannot be used in version 3.x resource files. The bitmap must be redrawn or converted to 3.x format.

Resource Compiler Error RC2171

unknown DIB header format

The bitmap header is not a BITMAPCOREHEADER or BITMAPINFOHEADER structure.

Resource Compiler Error RC2174**bitmap file filename is not in 2.03 format**

A bitmap used a format earlier than version 2.03. The bitmap must be converted or redrawn using the format for version 3.00 or later.

Resource Compiler Error RC2175**resource file filename is not in 3.00 format**

The specified resource used a format earlier than version 3.00. The resource file must be converted or recreated using the format for version 3.00 or later.

Resource Compiler Error RC2176**old DIB in filename.**

An old format Device Independent Bitmap was found in the specified file. It should be converted to the current format.

Resource Compiler Error RC2177**constant too big**

A constant value was too large to be represented in the type to which it was assigned.

Resource Compiler Error RC2180**unable to open temporary file**

The Resource Compiler was unable to open a temporary file. The probable cause is either that you do not have write permissions for the directory, or that the directory does not exist. The Resource Compiler attempts to use these files in the directory specified by the TMP (or GetSystemTempDirectory) environment variable or the current directory if none is specified.

Resource Compiler Error RC2181**duplicate font ID fontID**

The specified font ID was already defined.

Resource Compiler Error RC2189**#error : error**

This error is used to display other error messages.

Resource Compiler Error RC2235**too many arguments supplied**

An expression contained more formal parameters than expected.

Resource Compiler Error RC2236**required parameter missing**

An expression contained fewer formal parameters than expected.

[Return to Main Resource Compiler Messages Page](#)

Resource Compiler Warnings RC4000 through RC4413

The following topics cover Resource Compiler Warnings RC4000 through RC4413:

Resource Compiler Warning RC4000

UNKNOWN WARNING

Note the circumstances of the warning, try to isolate the problem and create a reproducible test case, then consult the technical-support help file or the technical-support section in one of your manuals.

Resource Compiler Warning RC4002

too many actual parameters for macro '*identifier*'

The number of actual parameters specified with the given identifier was greater than the number of formal parameters given in the macro definition of the identifier. The additional actual parameters were collected but ignored during expansion of the macro.

Resource Compiler Warning RC4003

not enough actual parameters for macro '*identifier*'

The number of actual parameters specified with the given identifier was less than the number of formal parameters given in the macro definition of the identifier. When a formal parameter is referenced in the definition and the corresponding actual parameter has not been provided, empty text is substituted in the macro expansion.

Resource Compiler Warning RC4004

missing close parenthesis after '*defined*'

An opening parenthesis was not matched with a closing parenthesis.

Resource Compiler Warning RC4005

'*identifier*' : macro redefinition

The given identifier was defined twice. The compiler used the second macro definition. This warning can be caused by defining a macro on the command line and in the code with a `#define` directive. It also can be caused by macros imported from include files. To eliminate the warning, either remove one of the definitions or use an `#undef` directive before the second definition.

Resource Compiler Warning RC4006

`#undef` expected an identifier

The name of the identifier whose definition was to be removed was not given with the `#undef` directive. The `#undef` directive was ignored.

Resource Compiler Warning RC4009

string too big, trailing chars truncated

A string was too large to fit in a buffer. Trailing characters were truncated and the remaining string was used.

Resource Compiler Warning RC4011

identifier truncated to '*identifier*'

An identifier was too long and was truncated to the name shown in the warning.

Resource Compiler Warning RC4012

float constant in a cross compilation

A float constant may not have the same value on different target platforms.

Resource Compiler Warning RC4067

unexpected characters following '*token*' directive - newline expected

A newline character should follow the specified token.

Resource Compiler Warning RC4079

unexpected token '*token*'

Check the syntax of the line containing the specified token.

Resource Compiler Warning RC4093

unescaped newline in character constant in inactive code

The constant expression of an `#if`, `#elif`, `#ifdef`, or `#ifndef` preprocessor directive evaluated to zero, making the code that follows inactive. Within that inactive code, a newline character appeared within a set of single or double quotation marks. All text until the next double quotation mark was considered to be within a character constant.

Resource Compiler Warning RC4111

unexpected token '*string*'

Check the syntax of the line containing the specified token.

Resource Compiler Warning RC4117

macro name '*macro*' is reserved, macro ignored

The specified macro was given a reserved name.

Resource Compiler Warning RC4203

SHIFT or CONTROL used without VIRTKEY

In an accelerator table resource, SHIFT or CONTROL require VIRTKEY. Because SHIFT and CONTROL are indicated as flag bits in a VIRTKEY type accelerator, they cannot exist independent from a VIRTKEY.

Resource Compiler Warning RC4204

ASCII character not equivalent to virtual key code

A string literal was used for the virtual key code in a VIRTKEY type accelerator.

This warning lets you continue, but be aware that the accelerator keys generated may not match the string you indicated. (VIRTKEYs use different key codes than ASCII accelerators.) While string literals are syntactically valid, you can only ensure that you get the accelerator you want by using the VK_* #define values in WINDOWS.H.

Resource Compiler Warning RC4205

string literal longer than 256 - stored anyway

A literal sting exceeded 256 characters. It was stored intact.

Resource Compiler Warning RC4206

name string too long - truncated at 256

A name string exceeded the limit of 256 characters. It was truncated to 256 characters.

Resource Compiler Warning RC4207

type string too long - truncated at 256

A type string exceeded the limit of 256 characters. It was truncated to 256 characters.

Resource Compiler Warning RC4208

title string too long - truncated at 256

A title string exceeded the limit of 256 characters. It was truncated to 256 characters.

Resource Compiler Warning RC4413

cannot determine file type for '*filename*' : assuming 8-bit ASCII

The type of specified file is not known to the Resource Compiler. It was compiled as an 8-bit ASCII file.

[Return to Main Resource Compiler Messages Page](#)

Resource Compiler Errors RW1004 through RW4004

The following topics cover Resource Compiler Errors RW1004 through RW4004:

Resource Compiler Fatal Error RW1004

Unexpected end of file

This error can be caused by omitting the linefeed and carriage-return characters on the last line of a text file.

Resource Compiler Fatal Error RW1009

Error creating resource-name

Could not create specified .RES file. Make sure it is not being created on a read-only drive. Use the /V option to find out whether the file is being created.

Resource Compiler Fatal Error RW1016

RC terminating after preprocessor errors

The Resource Compiler halted due to other errors. Fix the other errors and recompile.

Resource Compiler Fatal Error RW1021

I/O error reading file

The Resource Compiler was not able to read a file. Check that the drive containing the file is available and that the file is valid.

Resource Compiler Fatal Error RW1022

I/O error writing file

The Resource Compiler could not write to a file. This error may be caused by insufficient disk space. You must have free space equivalent to about twice the size of the executable file you are creating. Other possible reasons for this error are:

- Writing to a read-only volume
- Writing to a bad sector
- A sharing violation

Resource Compiler Fatal Error RW1023

I/O error writing file, drive full

The Resource Compiler could not write a file to a drive because the drive was full. You must have free space equivalent to about twice the size of the executable file you are creating.

Resource Compiler Fatal Error RW1025

Out of far heap memory

There wasn't enough memory. Consider using the Control Panel, System icon, Performance tab to increase your system's virtual memory, reboot, and retry.

Resource Compiler Fatal Error RW1030

Output Error

The output error can be caused by any of following conditions:

- Errors occurred when linking file
LINK failed. See Set Linker Options and Linker Reference for more information about linking an executable.
- .EXE file too large; relink with higher /ALIGN value
If the .EXE file is larger than 800K, you should use the /ALIGN:32 value on your LINK line.
- File not created by LINK
You must create the .EXE file with a version of Microsoft LINK later than 5.0 or with LINKW.
- Invalid .EXE file

Make sure that the linker created the .EXE file correctly and that the file exists. You can check the .EXE file with the DUMPBIN utility.

- Not a Microsoft Windows format .EXE file
Make sure that the linker created the .EXE file correctly and that the file exists. You can check the .EXE file with the DUMPBIN utility.
- Unable to create destination
RC was not able to create the destination file. Make sure that there is enough disk space and that you have write permission on the volume.
- Unable to open exe-file
RC could not open this .EXE file. Make sure that the linker created it correctly and that the file exists.

Resource Compiler Error RW2001

Invalid directive in preprocessed RC file

The RC file contains a `#pragma` directive. Use the `#ifndef` preprocessor directive with the `RC_INVOKED` constant that the Resource Compiler defines when it processes an include file. Place the `#pragma` directive inside a block of code that is not processed when the `RC_INVOKED` constant is defined. Code in the block is processed only by the C/C++ compiler and not by the Resource Compiler. The following sample code demonstrates this technique:

```
#ifndef RC_INVOKED
#pragma pack(2) // C/C++ only, ignored by Resource Compiler
#endif
```

The `#pragma` preprocessor directive has no meaning in an .RC file. The `#include` preprocessor directive is used frequently in an .RC file to include a header file (either a project-based custom header file or a standard header file provided by Microsoft with one of its products). Some of these include files contain the `#pragma` directive. Because a header file can include one or more other header files, the file that contains the offending `#pragma` directive may not be immediately obvious. The `#ifndef RC_INVOKED` technique can control including header files in project-based header files.

Resource Compiler Error RW2002

Parsing error

The parsing error can be caused by any of following conditions:

- Accelerator type required (ASCII or VIRTKEY)
The type field in the ACCELERATORS statement must contain either the ASCII or VIRTKEY value.
- BEGIN expected in accelerator table
The BEGIN keyword must immediately follow the ACCELERATORS keyword.
- BEGIN expected in dialog
The BEGIN keyword must immediately follow the DIALOG keyword.
- BEGIN expected in menu
The BEGIN keyword must immediately follow the MENU keyword.

- BEGIN expected in RCData
The BEGIN keyword must immediately follow the RCData keyword.
- BEGIN keyword expected in string table
The BEGIN keyword must immediately follow the STRINGTABLE keyword.
- Cannot re-use string constants
You are using the same value twice in a STRINGTABLE statement. Make sure you are not mixing overlapping decimal and hexadecimal values. Each ID in a STRINGTABLE must be unique. For maximum efficiency use contiguous constants that start on a multiple of 16.
- Control character out of range [^A - ^Z]
A control character in the ACCELERATORS statement is invalid. The character following the caret (^) must be between A and Z, inclusive.
- Empty menus not allowed
An END keyword appears before any menu items are defined in the MENU statement. Empty menus are not permitted by the Resource Compiler. Make sure you do not have any open quotation marks within the MENU statement.
- END expected in dialog
The END keyword must occur at the end of a DIALOG statement. Make sure there are no open quotes left from the preceding statement.
- END expected in menu
The END keyword must come at the end of a MENU statement. Make sure you do not have any open quotation marks or a mismatched pair of BEGIN and END statements.
- Expected comma in accelerator Table
RC requires a comma between the event and idvalue fields in the ACCELERATORS statement.
- Expected control class name
The class field of a CONTROL statement in the DIALOG statement must be one of the following types: BUTTON, COMBOBOX, EDIT, LISTBOX, SCROLLBAR, STATIC, or user-defined. Make sure the class is spelled correctly.
- Expected font face name
The typeface field of the FONT option in the DIALOG statement must be an ASCII character string enclosed in double quotation marks. This field specifies the name of a font.
- Expected ID value for menuitem
The MENU statement must contain a menuID field, which specifies the name or number that identifies the menu resource.
- Expected menu string
Each MENUITEM and POPUP statement must contain a text field, which is a string enclosed in double quotation marks that specifies the name of the menu item or pop-up menu. A MENUITEM SEPARATOR statement requires no quoted string.
- Expected numeric command value
RC was expecting a numeric idvalue field in the ACCELERATORS statement. Make sure that you have used a #define constant to specify the value and

- that the constant is spelled correctly.
- Expected numeric constant in string table
A numeric constant, defined in a #define statement, must immediately follow the BEGIN keyword in a STRINGTABLE statement.
 - Expected numeric point size
The pointsize field of the FONT option in the DIALOG statement must be an integer point size value.
 - Expected numerical dialog constant
A DIALOG statement requires integer values for the x, y, width, and height fields. Make sure that these values are included after the DIALOG keyword and that they are not negative.
 - Expected string in STRINGTABLE
A string is expected after each stringid value in a STRINGTABLE statement.
 - Expected string or constant accelerator command
RC was not able to determine what kind of key is being set up for the accelerator. The event field in the ACCELERATORS statement might be invalid.
 - Expecting number for ID
Expecting a number for the id field of a control statement in the DIALOG statement. Make sure you have a number or #define statement for the control ID.
 - Expecting quoted string in dialog class
The class field of the CLASS option in the DIALOG statement must be an integer or a string, enclosed in double quotation marks.
 - Expecting quoted string in dialog title
The captiontext field of the CAPTION option in the DIALOG statement must be an ASCII character string enclosed in double quotation marks.
 - File not found: filename
The file specified in the RC command line was not found. Check to see whether the file has been moved to another directory and whether the filename or path is typed correctly. Files are searched for using the INCLUDE environment variable or the Visual Workbench setting, if available.
 - Font names must be ordinals
The pointsize field in the FONT statement must be an integer, not a string.
 - Invalid accelerator
An event field in the ACCELERATORS statement was not recognized or was more than two characters in length.
 - Invalid accelerator type (ASCII or VIRTKEY)
The type field in the ACCELERATORS statement must contain either the ASCII or VIRTKEY value.
 - Invalid control character
A control character in the ACCELERATORS statement is invalid. A valid control character consists of one letter (only) following a caret (^).
 - Invalid control type
Each control statement in a DIALOG statement must be one of the following: CHECKBOX, COMBOBOX, CONTROL, CTEXT, DEFPUSHBUTTON,

EDITTEXT, GROUPBOX, ICON, LISTBOX, LTEXT, PUSHBUTTON, RADIOBUTTON, RTEXT, SCROLLBAR. Make sure these control statements are spelled correctly.

- Invalid type
The resource type was not among the types defined in the WINDOWS.H file.
- Text string or ordinal expected in control
The text field of a CONTROL statement in the DIALOG statement must be either a text string or an ordinal reference to the type of control is expected. If using an ordinal, make sure that you have a #define statement for the control.
- Mismatched parentheses
Make sure you have closed every open parenthesis in the DIALOG statement.
- Unexpected value in RCData
The raw-data values in the RCData statement must be integers or strings, each separated by a comma. Make sure you did not leave out a comma or leave out a quotation mark around a string.
- Unknown menu subtype
The item-definition field of the MENU statement can contain only MENUITEM and POPUP statements.

Resource Compiler Error RW2003

Generation Error

General errors can be caused by any of following conditions:

- Error: Bitmap file resource-file is not in 3.00 format
Bitmaps using the Windows version 2.x format cannot be used in version 3.x resource files. The bitmap must be redrawn or converted to 3.x format.
- Error: Old DIB in resource-name.
A Device Independent Bitmap (DIB) in the specified resource is not compatible with the 3.0 format. The bitmap must be redrawn or converted to the 3.x format.
- Error: Resource file resource-name is not in 3.00 format
An icon or cursor in the specified resource used a previous format. The icon or cursor must be redrawn or converted to the 3.x format.
- Unknown DIB header format
The bitmap header is not a BITMAPCOREHEADER or BITMAPINFOHEADER structure.
- Bitmap file resource file is not in 2.03 format
A bitmap used a format earlier than version 2.03. The bitmap must be converted or redrawn using the format for version 3.00 or later.
- Resource file is not in 3.00 format
A cursor or icon used a format earlier than version 3.00. The resource must be converted or redrawn using the format for version 3.00 or later.
- Unable to open temporary file
The Resource Compiler was unable to open a temporary file. The probable

cause is either that you do not have write permissions for the directory or that the directory does not exist. The Resource Compiler attempts to use these files in the directory specified by the TMP environment variable or the current directory if none is specified.

Resource Compiler Warning RW4001

[.EXE processing options \(/L /M /P /T /K /E /31 or /30\)](#)

.EXE processing options (/L /M /P /T /K /E /31 or /30) were specified when no executable file would be processed. These options can only be used when the result is an .EXE file (not a .RES file).

Resource Compiler Warning RW4003

[SHIFT or CONTROL used without VIRTKEY](#)

In an accelerator table resource, SHIFT or CONTROL require VIRTKEY. Because SHIFT and CONTROL are indicated as flag bits in a VIRTKEY type accelerator, they cannot exist independent from a VIRTKEY.

Resource Compiler Warning RW4004

[ASCII character not equivalent to virtual key code](#)

A string literal was used for the virtual key code in a VIRTKEY type accelerator. This warning lets you continue, but be aware that the accelerator keys generated may not match the string you indicated. (VIRTKEYs use different key codes than ASCII accelerators.)

[Return to Main Resource Compiler Messages Page](#)