pdp11

RT-11

System Message Manual

Order No. DEC-11-ORMEA-A-D

digital

RT-11

System Message Manual

Order No. DEC-11-ORMEA-A-D

ABSTRACT

This manual is for all RT-11 users. It provides a summary of error conditions that may occur during system use, along with recommended recovery procedures.

digital equipment corporation · maynard. massachusetts

First Printing, January 1976

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

PREFACE

The RT-11 System Message Manual summarizes all messages that may occur under the RT-11 Operating System, including RT-11 FORTRAN and BASIC/RT-11. It does not provide explanations for any message produced by other RT-11 application programs or systems; specific documentation for these applications must be consulted.

Three special categories of error conditions--hard errors, insufficient memory or storage space, and system failures--are described at the beginning of the manual. Users should read these sections in their entirety and then use the remainder of the manual for reference purposes.

It is suggested that this manual be left at the machine for use as a reference guide to all messages that may appear during system use.

1.0 HARD ERROR CONDITIONS

Special error conditions, called "hard" errors, are reported by the hardware rather than the system software. RT-11 interprets hard errors by printing a short message on the console terminal.

Often a hard error condition is simple to correct; the message will indicate an off-line or write-locked device, for example, and the system may "hang" (that is, wait) while the user corrects the problem, and then continue the operation.

Other hard errors indicate more serious problems such as bad blocks on the system device, or a malfunction in the hardware itself. It may be necessary for the user to reinitiate the operation or use another device; in extreme cases, the advice of a hardware expert may be called for.

The corrective action taken by the user depends upon the type of hard error condition and the device in use.

Following are the devices supported by RT-11 and the hard error conditions that may occur for each device. Corrective actions follow the error descriptions.

In all cases, if the error persists after all possible corrective actions have been tried, run the appropriate diagnostics and, if necessary, request the services of a field service representative.

1.1 Console Terminal Devices

Error Conditions:

Terminal devices do not report any hard error conditions. However, before using the system, ensure that the terminal is turned on, is on-line, and, if appropriate, has sufficient paper and a ribbon in good condition. The scope on a video terminal should be adjusted bright enough to be easily read.

Corrective Actions:

None.

1.2 Display Processor

Error Conditions:

The VT-11 display processor does not report any hard error conditions. Ensure that the display is turned on and the screen is adjusted bright enough to be easily read.

Corrective Actions:

None.

1.3 High-speed Paper Tape Devices

Error Conditions:

Reader: no tape; no power; off-line
Punch: no tape; no power

Corrective Actions:

Reader: ensure that the tape is properly positionec in the reader (over the read head with the tape retainer cover clcsed); check that the reader is turned on and set on-line.

Punch: ensure that the punch has been loaded properly with sufficient blank tape and is turned on, ready to punch.

1.4 Line Printers

Error Conditions:

No power; no paper; printer drum gate open; off-line; over temperature alarm; no printer connected to control unit (LV11).

Corrective Actions:

Ensure that the printer is turned on and set on-line; check that it is loaded properly with sufficient paper; ensure that a line printer is connected properly to the controller (LV11).

The system waits while corrective action is taken. (The monitor commands SET LP HANG/NOHANG may be used to control the wait feature; see Chapter 2 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D.)

1.5 Card Readers

Error Conditions:

No power. Indicator lights on the reader unit alert the user to these additional error conditions:

Read Check - card is torn on leading/trailing edges; card has punches in 0 or 81st column positions

Pick Check - card failed to move into the read station

Stack Check - previous card not properly seated in the output stacker (may be mutilated)

Hopper Check - (not present on all card readers) input hopper is empty; output stacker is full (occurs only during transfer initiation and only if SET CR NOHANG has been issued)

Corrective Actions:

Ensure that the card reader is turned on; check that the card deck is loaded properly; individual cards may need to be repunched.

The system waits while corrective action is taken. (The monitor commands SET CR HANG/NOHANG may be used to control the wait feature; see Chapter 2 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D.)

1.6 Cassettes

Error Conditions:

Write-lock; off-line; unit select; bad tape; block check (checksum error)

Corrective Actions:

Ensure that the write-protect tab on the cassette is properly positioned (the hole covered to write-enable the cassette, or uncovered to write-protect it); check that the cassette is correctly mounted on the proper drive (0-left, 1-right; system assumes 0 unless otherwise indicated).

For checksum and bad tape errors, retry the operation; use another cassette or try a different drive if possible, or use the PIP /G switch to ignore input errors while copying.

(The monitor commands SET CT RAW/NORAW may be used to control read-after-write error checking; see Chapter 2 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D.)

1.7 Magtapes

Error Conditions:

Cyclical redundancy or parity (checksum) error; bad tape; write-lock; off-line; unit select; power off

Corrective Actions:

Ensure that all magtape units are turned on (regardless of which is in use), set on-line and write-enabled, if appropriate (insert a write-ring on the back of a magtape to write-enable it); check that the tape is correctly mounted on the proper unit and that all units are assigned different select numbers; ensure that 7-track tapes are read on 7-track drives, and 9-track tapes on 9-track drives.

For checksum and bad tape errors, retry the operation; use another magtape or drive if possible, or use the PIP /G switch to ignore input errors while copying. Overwrite the last complete file on a bad tape so that the bad portion of the tape is within the area of the closed file.

1.8 DECtapes

Error Conditions:

Off-line; write-lock; unit select; parity (checksum) error; bad tape

Corrective Actions:

Ensure that the DECtape unit is set on-line and write-enabled, if appropriate; check that the tape is correctly mounted on the proper unit and that all units are assigned different select numbers.

For checksum and bad tape errors, retry the operation; use another DECtape or drive if possible, or use the PIP /G switch to ignore input errors while copying. Use the PIP /K switch to detect bad blocks; if the bad blocks fall within a file that is small, rename it with a .BAD extension; or use the PIP /T switch to create a file that encloses the bad blocks (see Chapter 4 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D, for detailed recovery instructions).

1.9 Disks

Error Conditions:

Off-line; write-lock; unit select; parity error; bad blocks; drive not ready

Corrective Actions:

Ensure that the disk drive is set on-line and write-enabled, if appropriate; check that the disk is correctly loaded in the proper unit, and that all units are assigned different select numbers. Ensure that a new disk is properly formatted before it is used.

For parity errors, retry the operation; use another disk or drive if possible, or use the PIP /G switch to ignore input errors while copying. Use the PIP /K switch to detect bad blocks; if the bad blocks fall within a file that is small, rename it with a .BAD extension; or use the PIP /T switch to create a file that encloses the bad blocks (see Chapter 4 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D, for detailed recovery instructions).

2.0 SYSTEM FAILURES

An RT-11 system failure occurs whenever the currently running program stops unexpectedly or suspends execution, leaving the system in what appears to be a nonfunctioning state. This section should aid the user in determining the cause of the system failure, and also in distinguishing between user errors and system errors.

Most system failures fall into one of the three categories: those that cause a return to the keyboard monitor, those that cause a monitor halt, and those that result in a program loop. Each is explained in detail below. While attempting to analyze a system failure, always keep in mind any new or unusual system features, such as user-written device handlers, a complex application program, or a special-purpose device. In addition, note the following items (the first five are generally important, and others may prove useful in particular cases):

1. Did any message appear on the terminal?

2. Is the processor halted or looping; what is the value of the program counter in either case?

3. If the processor is looping, do characters echo on the console; does CTRL C have any effect?

4. What are the contents of location 54 (which points to the base of RMON) and location 46 (which is the USR load address)? The location of the halt or the loop may be determined by comparing the value of the program counter with

these numbers. If the program counter is higher than these numbers, the half or loop occurred in the monitor and the difference between the numbers gives the offset into the monitor code.

5. What is the value of the stack pointer and the first several elements in the stack? Has the stack overflowed (that is, is the stack pointer <400)?

6. What are the contents of the registers?

7. Has monitor code been corrupted? Determine from a source listing, if available, the integrity of significant areas in RMON, especially the area immediately below the monitor stack.

8. Determine from a source listing, if available, what code is indicated by the halt or loop.

9. What are the contents of the monitor data base, as appropriate (for example, the address of the running job or the addresses of the loaded handlers)?

10. Can the problem be localized to a single job? If so, what are the contents of the job impure area (the job status word, channel status words, the queue elements)?

11. Can the problem be localized to a single device? If so, what are the contents of the handler data and queue and the device status registers?

The remainder of this section makes reference to information contained in the RT-11 Software Support Manual (DEC-11-ORPGA-B-D) and in the monitor listings.

2.1 Return to Keyboard Monitor

A return to the keyboard monitor is recognized when the monitor dot is printed at the left margin of the console terminal. If a monitor message (one beginning with ?M-) has also been printed, refer to its meaning and corrective action as listed in the system message section of this manual (Section 4.0).

If no message is printed (only the monitor dot), an interrupt through an empty vector to code at location 0 is indicated:

```
LOC.
 0       BIC R0,R0     ;TO ENSURE A HARD EXIT
 2       .EXIT         ;BACK TO KMON
```

Possible causes of this error include the following: a spurious interrupt request appeared, a vector was never filled, a filled vector was not protected under the F/B monitor, an .ASECT was attempted into a protected vector, or an applications program was run with an incorrect vector address.

2.2 Monitor Halts

Monitor halts are easily recognized by the fact that they occur in high memory above the contents of location 54 (the RMON base address pointer). When a monitor halt occurs, do not attempt to restart the system by pressing CONTinue on the processor; the system must be rebooted.

2.2.1 Single-Job Monitor Halts

The Single-Job Monitor has five potential halts. The first halt is in absolute location 26 and occurs on power fail. This vector is filled by the system bootstrap and protected by the permanent bitmap.

The second halt is at location 116 and occurs on a memory parity error. This location is filled by the system bootstrap and protected by the permanent bitmap.

The next halt is 1426 octal bytes into RMON (offset from the contents of location 54) and occurs on stack overflow. The top of the stack contains the location of the instruction causing the stack overflow. Stack overflow is a complex error condition, and is discussed in detail later.

The fourth halt is 5164 octal bytes into RMON (offset from the contents of location 54) and is executed if a hardware error occurs while swapping part of the user program to the monitor swap blocks or while reading the KMON or USR from the system device. This halt indicates a temporary or permanent failure in the controller or the physical volume. The most common cause is a write-locked system device.

The fifth halt is 6060 octal bytes into RMON (offset from the contents of location 54) and indicates that a hard I/O error appeared and that the user set bit 7 of the job status word (absolute location 44) to request a halt on hard error. Register 5 points to the queue element, specifically to the pointer to the channel status word; register 4 points to the handler, specifically to the current queue element pointer. Examine the system device status registers for more information.

Any other halts that may occur indicate that the Single-Job Monitor code itself has been corrupted.

2.2.2 Foreground/Background Monitor Halts

The Foreground/Background Monitor has three halts. The first is the power fail halt, identical to the power fail halt in the Single-Job Monitor.

The second is the memory parity error halt, identical to the memory parity error halt in the Single-Job Monitor.

The third halt is 1652 octal bytes into RMON (offset from the contents of location 54) and is reached on a trap to 4 or trap to 10 while the monitor is in system state (that is, while the monitor is processing a real or fake interrupt -- executing code in interrupt service routines, device handlers, or selected portions of the monitor). The address where the trap occurred is at the top of the stack. If this address is within user code, investigate an error in an interrupt service routine or device handler. Verify that handlers are not .FETCHed into areas that will be destroyed by data buffers or overlayed by USR swapping (see below). If a nonexistent device was called, the handler traps when the device registers are referenced. (This is best avoided by deleting from the system disk all handlers or devices that are not part of the system configuration.) If the address is in the monitor, subtract the contents of location 54 and examine the monitor code. The monitor code may be corrupted, or data in the monitor or user region (such as queue elements or channel status tables) may be corrupted. Hardware problems causing bus timeout will trap through location 4, but this is extremely rare and should be investigated as a last resort. Absolute location 52 is negative on a trap to 10, and 0 or positive on a trap to 4.

2.2.3 USR Swapping - Many system failures are caused by the USR swapping over important memory areas (such as device handlers, queue elements, completion routines; this may occur when running FORTRAN programs that use SYSLIB calls). One way to detect this type of failure is to SET USR NOSWAP and rerun the program (providing enough free memory exists). If the failure does not recur, then USR swapping is probably causing the problem. The program should be changed such that the USR does not swap over it at all (by being linked with overlays or with a different bottom address) or by ensuring that the USR does not swap over any important areas within the program. See Section 9.2.5 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D, for details concerning the swapping algorithm.

2.2.4 Stack Overflow - Stack overflow occurs when the stack is pushed through its low limit. It may or may not be detected depending on the location of the stack. The normal location for the user background stack is 1000, with a low limit of 400. Most PDP-11 processors detect stack overflow at 400 and generate a trap to 4. (Some of the new processors, for example, the PDP-11/03, do not support this feature.) If the stack is located elsewhere, overflow detection is not supported by the RT-11 system.

Stack overflow is typically a fatal condition. RT-11 treats all detected user stack overflows as fatal and aborts the offending program. Under the Single-Job Monitor, the system halts. The F/B Monitor, which has less severe size constraints, aborts the program with the ?M-TRAP TO 4 message.

The FORTRAN Object Time System is a frequent cause of user stack overflow, since it uses large amounts of stack. Extra stack can be allocated for background jobs by using the Linker /B switch to raise the program base. Extra stack can be allocated for foreground jobs at runtime by using the FRUN /S switch.

Monitor stack overflow will generally not occur, since enough stack space is allocated to handle "worst-case" situations.

In the case of a user-written device handler that requires a large amount of stack, it is recommended that stack space be allocated within the handler and that a register other than SP be used to reference the stack.

2.3 Program Looping

Following are some of the possible loops that may be entered in the Single-Job and F/B Monitors. If the system seems to be "hung" in a loop (that is, executing a set of instructions infinitely) step through the loop to determine where in memory the loop occurs.

If the system contains a KW11L clock, or some otler device that generates frequent interrupts, first disable this device, if possible, by clearing the interrupt enable bit in the appropriate device status register. It is then possible to single-step through the loop by first pressing HALT, and then pressing the CONTINUE key (on the computer console) while the HALT key remains pressed.

If the device connot be disabled, it is necessary to alternately press/raise HALT and press CONTINUE, thus examining random locations in the loop. (Otherwise, single-stepping will follow the interrupt service routine of the device, and not the desired program loop.)

2.3.1 Single-job Monitor Looping - At 2150 octal bytes into RMON (offset from the contents of location 54) is a loop that waits for all I/O to complete before executing an exit or a chain. When the system becomes caught in this loop, examine the queue elements and channel status word to find the outstanding I/O requests. Examine the appropriate device to determine why it is not satisfying the request.

At 5660 octal bytes into RMON (offset from the contents of location 54) is a loop that waits for a queue element. An active program that remains in this loop for long periods of time is bound by queue elements and should have its number of queue elements expanded. A program that hangs in this loop indefinitely indicates that a device is not satisfying an I/O request. Check the queue elements and channel status tables to determine the responsible device.

At 5644 octal bytes into RMON (offset from the contents of location 54) is a loop that waits for I/O completion on a particular channel. Register 3 points to the channel status table for that channel. Hanging in this loop is an indication that a device is not satisfying an I/O request.

Loops occurring at 514 octal and 5452 octal bytes into the USR (offset from the contents of location 46 or of the location that is 266 octal bytes into RMON) serve the purpose of delaying execution until all I/O is finished. The first is reached by a channel define programmed request and the second by a soft reset. If a program gets caught in this loop, examine the queues, channel tables, and device queues to locate the bad device; then examine the device registers.

At 5352 octal bytes into the USR (offset from the contents of location 46 or 266 octal bytes into RMON) is a loop that delays a hard reset until console terminal output is done. If the system seems to hang in this loop, verify that the console terminal vectors are still pointing into RMON, that the output interrupt is enabled, and that the console is operative. Static electricity is frequently responsible for clearing the terminal interrupt enable bit; check the humidity conditions in the room.

At 4722 octal bytes into RMON (offset from the contents of location 54) is a loop that waits for I/O completion on a given channel and is entered when a .WAIT request is issued. Register 3 points to the channel status table, from which the offending device can be determined.

2.3.2 F/B Monitor Looping - At 5022 octal bytes into RMON (offset from the contents of location 54) is a loop that is entered after the ?M-SYS ERR message has been generated, due to a hard error on reading the KMON and USR. A controller malfunction is indicated, but the loop continues to try the read operation. Halt the processor. The system may not boot; check the system device for bad blocks.

At 13506 octal bytes into RMON (offset from the contents of location 54) is a scheduler and null job loop, indicating that neither the background nor foreground job is runnable. When the system remains in this loop, check the impure areas of the two jobs to locate a cause. Two pointers are useful for this purpose. One is at 1504 octal bytes into RMON and is a pointer to the background impure area; the other is the pointer to the foreground impure area and is at 1506 octal bytes into RMON. Using these pointers, the I.JSTA words (see the RT-11 Software Support Manual, DEC-11-ORPGA-B-D), the channel tables, and the queues may be examined to determine why neither job is runnable. Note that a job can be "blocked" by a lack of available queue elements, although I.JSTA is zero. On every significant event, the job will be run to check for an available queue element. Note also that the $EXIT bit in I.JSTA can be set in many ways--through the .EXIT, CDFN, or .SRESET programmed requests, for example.

2.3.3 Handler Looping - By default, the card reader and line printer handlers will loop on hardware errors until the error condition has been removed. The system will loop through the monitor service code and the handler detection code. This condition can be changed using the monitor SET command (see the RT-11 System Reference Manual, DEC-11-ORUGA-C-D). The other supplied handlers use counts to prevent indefinite retries on a device.

3.0 INCREASING MEMORY AND STORAGE SPACE

Some RT-11 System errors are caused because there is insufficient free memory space to accomodate a particular operation or insufficient space for an output file on a storage device. Using one (or more) of the procedures listed below may suffice to eliminate the problem.

During an output operation to an RT-11 directory-structured device, an error message may indicate insufficient room in the directory or on the device. Try the following to create more space:

Delete unnesessary files from the output device, perhaps transferring them to a backup device;

Use another device with more space;

During a PIP transfer, use the /X option to place the file in the first free area large enough to accomodate it (RT-11 otherwise uses one half the largest available space for an output file; see the .ENTER request in Chapter 9 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D);

Specify an explicit output file size using the [n] construction;

Compress the device using the PIP /S option; this creates the largest possible empty space on the device by condensing all the free blocks into one area, and also makes more efficient use of the directory space;

If directory overflow persists, allocate more directory segments on another volume (using the PIP /Z/N:n switches) and transfer the files to this volume, or use the DIRECT program described in Chapter 4 of the RT-11 System Generation Manual (DEC-11-ORGMA-A-D).

If device overflow persists, upgrade to a larger volume (for example, from RX11 to RK11, or from RK11 TO RPR02).

When a program is loaded or during program execution, an error message may indicate insufficient room in main memory. The following methods make more memory available by redesigning the program:

Use single-buffering instead of double-buffering;

Use smaller I/O buffers;

Decrease the maximum number of channels open simultaneously;

Break the code into smaller modules for more efficient overlaying;

Remove any testing code no longer required;

Use algorithms which require less main memory;

Transfer more data storage to the mass storage devices;

Break the program into several programs, to permit chaining between them.

The following methods make more memory available without redesigning the program:

Remove any unnecessary loaded handlers, including the BATCH handler and the scroller (GT) handler;

Overlay the program;

Unload the foreground program;

Use the Single-Job Monitor (it requires approximately one half the space needed by the Foreground/Background Monitor);

Use the SET USR SWAP command (see Chapter 2 of the RT-11 System Reference Manual, DEC-11-ORUGA-C-D) to allow USR swapping.

If none of these methods alleviate the space problem, the application may not be suitable for the current system.

4.0 SYSTEM MESSAGES

Each RT-11 system message is shown in this section in italics. Under the message appear four columns. The first column lists those modules (i.e., system programs) that can produce the message. The second column indicates the type of condition according to the following categories:

Abbreviation	Type	Meaning
A	ACTION	Message is sent to the terminal and execution stops. A user-response is expected before execution can continue.
C:n	COUNT:n	Message is sent to the terminal; execution continues until the nth occurrence of the error, at which time it is treated as FATAL (for FORTRAN messages only).
F	FATAL	Message is sent to the terminal or the listing output file; the current command or statement is ignored, or execution is terminated, in which case the user must enter another command or correct the statement in error.
IG	IGNORE	A subcategory of INFORMational, used for FORTRAN messages only. When an error in this category occurs, execution continues. No message is sent to the terminal or listing output file unless the user has (optionally) changed its type classification by a call to the FORTRAN error routine, SETERR. See the RT-11/RSTS/E FORTRAN IV User's Guide (DEC-11-LRRUA-A-D) for instructions.
IN	INFORM	Informational message; an error condition is detected and the user is informed, either at the terminal or in the listing file; execution continues. The error condition may affect execution at a later time and may require future action.
W	WARNING	A subcategory of INFORMational; message is sent to the terminal or the listing output file, and execution continues totally.

The third column gives the meaning of the message. The final column describes possible recovery procedures (if any) by the system, the user, or both.

Messages preceded by a numeric value are listed first (in numerical order); all others follow in alphabetical order, with special characters and spaces preceding alphabetics in the ordering.

Ø *NON-FORTRAN ERROR CALL*

FORTRAN	F or IN	This message indicates an error condition (not internal to the FORTRAN run-time system) that may have been caused by one of four situations:	
		A foreground job using SYSLIB completion routines was not allocated enough space (using the FRUN /N switch) for the initial call to a completion routine.	Check Appendix G (Section G.1) and Appendix O (Section O.1.4) of the <u>RT-11 System Reference Manual</u>, DEC-11-ORUGA-C-D, for the formula used to allocate more space.
		There was not sufficient memory for the background job.	Refer to Section 3.0, at the beginning of this manual for information on how to increase memory space.
		Under the Single-Job Monitor, a SYSLIB completion routine interrupted another completion routine.	Use the F/B Monitor to allow more than one active completion routine (see Appendix O of the <u>RT-11 System Reference Manual</u>, DEC-11-ORUGA-C-D).
		An assembly language module linked with a FORTRAN program issued a TRAP instruction with an error code that was not recognized by the FORTRAN error handler. (Note that error messages produced by the FORTRAN extensions package contain this message, preceded by a line describing the error in more detail.)	Correct the program logic.

1 *INTEGER OVERFLOW*

FORTRAN	F	During an integer multiplication, division, or exponentiation operation, the value of the result exceeded 32767 in magnitude.	Correct the program logic.

2 INTEGER ZERO DIVIDE

FORTRAN	F	During an integer mode arithmetic operation, an attempt was made to divide by zero.	Correct the program logic.

3 COMPILER GENERATED ERROR

FORTRAN	F	An attempt was made to execute a FORTRAN statement in which the compiler had previously detected errors.	Consult the program listing generated by the compiler (if one was requested) and correct the program for the errors generated at compile-time.

4 COMPUTED GO TO OUT OF RANGE

FORTRAN	W	The integer variable or expression in a computed GO TO statement was less than 1 or greater than the number of statement label references in the list.	Control is passed to the next executable statement. Examine the source program and correct the program logic.

5 INPUT CONVERSION ERROR

FORTRAN	C:3	During a formatted input operation, an illegal character was detected in an input field.	A value of 0 is returned. Examine the input data and correct the invalid record or the program logic.

6 OUTPUT CONVERSION ERROR

FORTRAN	IG	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.	The field is filled with *'s. Correct the FORMAT statement to allow a greater field length.

1Ø FLOATING OVERFLOW

FORTRAN	C:3	During an arithmetic operation, a real value has exceeded the largest representable real number.	A value of zero is returned. Correct the program logic.

11 FLOATING UNDERFLOW

FORTRAN	IG	During an arithmetic operation, a real value has become less than the smallest representable real number.	The real number is replaced with a value of zero. Correct the program logic.

12 FLOATING ZERO DIVIDE

FORTRAN	F	During a real mode arithmetic operation an attempt was made to divide by zero.	Correct the program logic.

13 SQRT OF NEGATIVE NUMBER

FORTRAN	C:3	An attempt was made to take the square root of a negative number.	The result is replaced by zero. Correct the program logic.

14 UNDEFINED EXPONENTIATION OPERATION

FORTRAN	F	An attempt was made to perform an illegal exponentiation operation. (For example, -3.**.5 is illegal because the result would be an imaginary number.)	Correct the program logic.

15 LOG OF NEGATIVE NUMBER

FORTRAN	F	An attempt was made to take the logarithm of a negative number.	Correct the program logic.

16 WRONG NUMBER OF ARGUMENTS

FORTRAN	F	One of the FORTRAN Library functions, or one of the System Subroutines which checks for such an occurrence, was called with an improper number of arguments.	Check the format of the particular library function or System Subroutine call, and correct the call.

2Ø INVALID CHANNEL NUMBER

FORTRAN	F	An illegal logical unit number was specified in an I/O statement.	A logical unit number must be an integer within the range 1 to 99. Correct the statement in error.

21 NO AVAILABLE CHANNELS

FORTRAN	F	An attempt was made to have too many devices simultaneously open for I/O.	The maximum number of active channels is six by default. To increase the maximum, recompile the main program using the /N switch to specify a larger number of available channels (the legal range is 1 to 15).

22 INPUT RECORD TOO LONG

FORTRAN	F	During an input operation, a record was encountered that was longer than the maximum record length.	The default maximum record length is 136 (decimal) bytes. To increase the maximum, recompile the main program using the /R switch to specify a larger run-time record buffer (the legal range is 4 to 4095).

23 HARDWARE I/O ERROR

FORTRAN	F	A hardware error was detected during an I/O operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

24 ATTEMPT TO READ/WRITE PAST END OF FILE

FORTRAN	F	During a sequential write operation, this message indicates that the space allocated to the file was insufficient; during a sequential read, an attempt was made to read beyond the last record of the file. During a random-access write operation, this message indicates that the space allocated to the file was insufficient, or that a programming error occurred during a read (such as attempting to reference a record number that was not within the bounds of the file).	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space. Use an END= parameter on a sequential read. Correct the progamming logic error in a random-access read operation. Use the square bracket construction to make more space available when opening a file with CALL ASSIGN.

25 ATTEMPT TO READ AFTER WRITE

FORTRAN	F	An attempt was made to read after writing on a file.	A write operation must be followed by a REWIND or BACKSPACE before a read operation can be performed. Correct the program logic.

26 RECURSIVE I/O NOT ALLOWED

FORTRAN	F	An expression in the I/O list of a WRITE statement caused initiation of another READ or WRITE operation. (This can happen if a FUNCTION that performs I/O is referenced in an expression in a WRITE statement I/O list.)	Correct the program logic.

27 ATTEMPT TO USE DEVICE NOT IN SYSTEM

FORTRAN	F	An attempt was made to access a device that was not legal for the system in use.	Reassign the device to a legal device, or change the statement in error.

28 OPEN FAILED FOR FILE

FORTRAN	F	The file specified was not found, there was no room on the device, or FORTRAN selected a channel already in use.	Verify that the filename exists as specified. Verify that FORTRAN default unit numbers are assigned to the devices expected. Verify that the channel selected is not already in use by an assembler-level or SYSLIB call. Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

29 NO ROOM FOR DEVICE HANDLER

FORTRAN	F	There was not enough free memory left to accommodate a specific device handler.	Move the file to the system device or to a device whose handler is resident; refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Recompile the FORTRAN program with /S or link with $SHORT. Use /N to reduce the number of logical units or /R to reduce the record size buffer.

3Ø NO ROOM FOR BUFFERS

FORTRAN	F	There was not enough free memory left to set up required I/O buffers.	Reduce the number of logical units that are open simultaneously at the time of the error (/N); refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Recompile the FORTRAN program with /S or link with $SHORT. Use /R to reduce the record size buffer.

31 NO AVAILABLE RT-11 CHANNEL

FORTRAN	F	More than the maximum number of RT-11 channels available to the FORTRAN run-time system (15) were requested to be simultaneously opened for I/O.	Close any logial units previously opened that need not be open at this time.

32 FMTD-UNFMTD-RANDOM I/O TO SAME FILE

FORTRAN	F	An attempt was made to perform any combination of formatted, unformatted, or random-access I/O to the same file.	Correct the program logic.

33 ATTEMPT TO READ PAST END OF RECORD

FORTRAN	F	An attempt was made for unformatted I/O to read a larger record than actually existed in a file.	Check the construction of the data file; correct the program logic.

34 UNFMTD I/O TO TTY OR LPT

FORTRAN	F	An attempt was made to perform an unformatted write operation on the terminal or line printer.	Assign the logical unit in question to the appropriate device using the ASSIGN Keyboard Monitor command, the ASSIGN FORTRAN library routine, or the IASIGN SYSLIB routine.

35 ATTEMPT TO OUTPUT TO READ ONLY FILE

FORTRAN	F	An attempt was made to write on a file designated as read-only.	Check the CALL ASSIGN system subroutine or IASIGN SYSLIB function to ensure that the correct arguments were used. Check for a possible programming error.

36 BAD FILE SPECIFICATION STRING

FORTRAN	F	The Hollerith or literal string or array specifying the RT-11 device/filename in the CALL ASSIGN system subroutine could not be interpreted.	Check the format of the CALL ASSIGN statement. If the square bracket construction is used, an equal sign must follow it.

37 RANDOM ACCESS READ/WRITE BEFORE DEFINE FILE

FORTRAN	F	A random-access read or write operation was attempted before a DEFINE FILE was performed.	Correct the program so that the DEFINE FILE operation is before any random-access read or write operation.

38 RANDOM I/O NOT ALLOWED ON TTY OR LPT

FORTRAN	F	Random access I/O was illegally attempted on the terminal or line printer.	Assign the logical unit in question to the appropriate device using the ASSIGN Keyboard Monitor command, the ASSIGN FORTRAN library routine, or the IASIGN SYSLIB routine.

39 RECORD LARGER THAN RECORD SIZE IN DEFINE FILE

FORTRAN	F	A record was encountered that was larger than that specified in the DEFINE FILE statement for a random-access file.	Shorten the I/O list or redefine the file specifying larger records.

40 REQUEST FOR A BLOCK LARGER THAN 65535

FORTRAN	F	An attempt was made to reference an absolute disk block address greater than 65535.	Correct the program logic.

41 DEFINE FILE ATTEMPTED ON AN OPEN UNIT

FORTRAN	F	A file was open on a unit and another DEFINE FILE was attempted on that unit.	Close the open file using CALL CLOSE before attempting another DEFINE FILE.

42 MEMORY OVERFLOW COMPILING OBJECT TIME FORMAT

FORTRAN	F	The OTS ran out of free memory while scanning an array format generated at run-time.	Use a FORMAT statement specification at compile-time rather than object-time formatting. Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.

43 SYNTAX ERROR IN OBJECT TIME FORMAT

FORTRAN	F	A syntax error was encountered while the OTS was scanning an array format generated at run-time.	Correct the programming error.

44 2ND RECORD REQUEST IN ENCODE/DECODE

FORTRAN	F	An attempt was made to use ENCODE and DECODE on more than one record.	Correct the FORMAT statement associated with the ENCODE OR DECODE so that it specifies only one record. Verify that there is no '\' in the FORMAT statement and that unexpected format reversion does not occur (see the <u>PDP-11 FORTRAN Language Reference Manual</u>, DEC-11-LFLRA-B-D)

45 INCOMPATIBLE VARIABLE AND FORMAT TYPES

FORTRAN	F	An attempt was made to output a real variable with an integer field descriptor or an integer variable with a real field descriptor.	Correct the FORMAT statement associated with the READ or WRITE, ENCODE or DECODE.

46 INFINITE FORMAT LOOP

FORTRAN	F	The format associated with an I/O statement that includes an I/O list had no field descriptors to use in transferring those variables.	Correct the FORMAT statement in error.

47 ATTEMPT TO STORE OUTSIDE PARTITION

FORTRAN	F	In an attempt to store data into a subscripted variable, the address calculated for the array element in question did not lie within the section of memory allocated to the job. The subscript in question was out-of-bounds. (This message is issued only when bounds-checking modules have been installed in FORLIB, i.e., FORLIB.V2S.)	Correct the program logic.

48 UNIT ALREADY OPEN

FORTRAN	F	An attempt was made to perform an operation illegal on an open file.	Close the file (using CALL CLOSE) before attempting to use the unit.

49 ENDFILE ON RANDOM FILE

FORTRAN	F	An ENDFILE statement specified a unit number of a file which was currently open as a random-access file. (ENDFILE applies only to sequential files.)	Correct the program logic.

59 USR NOT LOCKED

FORTRAN	W	This message is issued when the FORTRAN program is started if the program was running in the foreground, the /U switch was used during compilation, and the USR was swapping (i.e., a SET USR NOSWAP command has not been done).	Reexamine the intent of the /U switch at compile time and either compile without /U or issue a SET USR NOSWAP command before running the program.

6Ø STACK OVERFLOWED

FORTRAN	F	The hardware stack overflowed. More stack space may be required for subprogram calls and opening of files. Proper traceback is impaired. This message occurs in the background only.	Allocate additional space by using the /B switch at link-time. Check for a programming error.

61 ILLEGAL MEMORY REFERENCE

FORTRAN	F	Some type of BUS error occurred, most probably an illegal memory address reference. (This is the FORTRAN equivalent of the Monitor ?M-TRAP TO 4 message.)	If the error occurred within a user-written assembly language routine, check for an error in the source code and correct the programming logic. If the error occurred in the FORTRAN extensions package, verify that the register addresses used in the extensions library correspond to those on the present hardware. Verify that the correct FORTRAN library is being used.

62 FORTRAN START FAIL

FORTRAN	F	The program was loaded into memory but there was not enough free memory remaining for the OTS to initialize work space and buffers.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Recompile the FORTRAN program with /S or link with $SHORT; if running a foreground job, specify a larger value using the FRUN /N switch. Refer to the formulas in Appendix G (Section G.1) and Appendix O (Section O.1.4) of the <u>RT-11 System Reference Manual</u> DEC-11-ORUGA-C-D.

63 ILLEGAL INSTRUCTION

FORTRAN	F	The program attempted to execute an illegal instruction (e.g., floating point arithmetic instruction on a machine with no floating	If the error occurred within a user-written assembly language routine, check for an error in the source code and correct the programming logic.

			point hardware). (This is the FORTRAN equivalent of the Monitor ?M-TRAP TO 10 message.)	Otherwise, verfiy that the correct FORTRAN library is being used.
******* A*	MACRO	IN	Addressing error. An address within the instruction was incorrect; an attempt was made to add two relocatable symbols, or a local symbol was defined more than 128 words from the beginning of a local symbol block.	These errors indicate unacceptable input format. The source program should be corrected and reassembled.
A	MACRO	IN	Addressing error. An address within the instruction was incorrect; an attempt was made to add two relocatable symbols, or a local symbol was defined more than 128 words from the beginning of a local symbol block.	These errors indicate unacceptable input format. The source program should be corrected and reassembled.
ABORT JOB	BATCH	F	Either an error has occurred in compiling a BATCH program, or a diagnostic compile was requested with /N.	The compiler forces the job to abort. Check the log file for all error messages.
ADDITIVE REF OF xxxxxx AT SEGMENT # yyyyyy	LINKER	W	Rule 1 of the overlay rules explained in Chapter 6, Section 6.6 of the RT-11 System Reference Manual (DEC-11-ORUGA-C-D) has been violated. xxxxxx represents the entry point; yyyyyy represents the segment number.	Ensure that calls or branches to overlay segments are made directly to entry points in the segment.

?ADDR?

MONITOR	F	An address was out of range in an E or D command.	The allowable range is between 0 and the base of RMON (contents of location 54 octal); the locations in the E or D commands should not exceed this range.

?ADDR NOT IN SEG?

PATCH	F	The address was not in the specified overlay segment.	Recheck the Linker load map for the address and proper overlay segment.

ADJUSTABLE DIMENSIONS ILLEGAL FOR ARRAY ****

FORTRAN	IN	An adjustable array was not a dummy argument in a subprogram, or the adjustable dimensions were not integer dummy arguments in the subprogram. (**** represents the array name.)	The compiler sets the dimension to 1. Correct the source program.

?filnam.ext ALREADY EXISTS?

FILEX	F	An attempt was made to create the named file (filnam.ext) on a DOS DECtape when a file already existed under the name specified.	Use /D to delete the file and retry the transfer, or choose a new name for the file to be created.

ARE YOU SURE?

CBUILD	A	Confirmation must be given by the user before the operation can be performed.	Type Y and a carriage return to allow the operation; type N and a carriage return to abort the operation.

?ARG AT LINE xxxxx

BASIC	F	Arguments in a function call did not match (in number or in type) the arguments defined for the function.	Check that the arguments to the function are of the proper type and number and are in the correct range.

ARGUMENT ERROR AT LINE xxxxx

BASIC	F	Arguments in a function call did not match (in number or in type) the arguments defined for the function.	Check that the arguments to the function are of the proper type and number and are in the correct range.

ARRAY [****] EXCEEDS MAXIMUM SIZE

FORTRAN	IN	The storage required for a single array or for all arrays in total was more than physically addressable (greater than 32K words). (****, if printed, represents the array name.)	Correct the statement in error or reduce the space necessary for array storage.

*ARRAY **** HAS TOO MANY DIMENSIONS*

FORTRAN	IN	An array had more than seven dimensions. (**** represents the array name.)	Correct the program. The legal range for dimensions is 1 to 7.

ARRAYS TOO LARGE AT LINE xxxxx

BASIC	F	There was not enough room in the memory available for the arrays specified in the DIM statements.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines; Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Reduce the size of arrays in memory to the size required (DIMension statement);

Use virtual array files for arrays that are too large to fit into memory;

Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays;

Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use.

After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

?ATL AT LINE xxxxx

BASIC   F   There was not enough room in the memory available for the arrays specified in the DIM statements.

Reduce the program size. This can be done by one of several procedures:

Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET;

Make maximum use of multiple statement lines;

Make efficient use of program loops, subroutines, and user-defined functions;

Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements;

Reduce the size of arrays in memory to the size required (DIMension statement);

			Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

ATTEMPT TO EXTEND COMMON BACKWARDS

FORTRAN	IN	While attempting to equivalence arrays in COMMON, an attempt was made to extend COMMON past the recognized beginning of COMMON storage.	Correct the program logic.

******* B*

FORTRAN	IN	Columns 1-5 of a continuation line were not blank.	Columns 1-5 of a continuation line must be blank except for a possible D' in column 1; the columns are ignored and compilation continues.
MACRO	IN	Bounding error. This error indicates that instructions or word data would be assembled at an odd address in memory.	The location counter is updated by +1. The source program should be corrected.

B				
	MACRO	IN	Bounding error. This error indicates that instructions or word data would be assembled at an odd address in memory.	The location counter is updated by +1. The source program should be corrected.
?/B NO VALUE?				
	LINKER	F	No argument was specified to the /B switch.	Reenter the command string specifying an unsigned even octal number as the argument to the /B switch.
?/B ODD VALUE?				
	LINKER	F	The argument to the /B switch was not an unsigned even octal number.	Reenter the command string specifying an unsigned even octal number as the argument to the /B switch.
?BAD BOOT?				
	PIP	F	The boot switch (/O) was specified on a nonfile-structured device.	Reenter the command indicating a legal device name.
?BAD CHECKSUM?				
	PATCHO	F	A formatted binary block in the input file had a checksum that did not agree with that calculated for its contents.	Ensure that the correct module is being patched. This error may also indicate that a patch has been made incorrectly to a preceding file, or that data has been lost in the input file. If the error persists, obtain a new copy of the .OBJ module.
BAD CONSTRUCTION				
	BATCH	F	In RT-11 mode, an IF statement was not in the correct form or there was an illegal 'text' directive in a command.	Verify that the format of the IF statement is correct, and that the 'text' directive is valid as entered.

BAD COPY OF HANDLER

BATCH	F	The copy of BA.SYS in memory is bad.	UNLOAD the copy of BA.SYS in memory, reload BA.SYS, and run BATCH.

BAD DATA READ AT LINE xxxxx

BASIC	F	Item input from a DATA statement list by a READ statement was bad.	Ensure that the elements in the data list are in the correct format.

BAD DATA-RETYPE FROM ERROR

BASIC	A	The item entered to an input statement was bad.	Retype the item that caused the error and execution will continue.

?BAD GSD?

LINKER	F	There was an error in the global symbol directory (GSD). The file is probably not a legal object module. This error message occurs on pass 1 of the Linker.	Verify that the correct filenames were specified as input; check for a typing error in the command line. Reassemble or recompile the source to obtain a good object module and retry the operation.

?BAD LIBR?

LIBR	F	An attempt was made to build a library file containing no directory entries or to use an illegally constructed library file as input to the Librarian.	Verfiy that the correct filenames were specified as input; check for a typing error in the command line. Verify that the input to the Library has at least one directory entry and is valid input. It may be necessary to rebuild the library file.

BAD MACRO ARG

EXPAND	IN	The macro argument was not formatted correctly.	The macro is ignored and execution continues. Correct the source program.

?BAD OBJ?

LIBR	F	A bad object module was detected during input.	Check for a typing error in the command line; verify that the correct files were specified as input. Reassemble or recompile to obtain a good object module and retry the operation.
PATCHO	F	The input file contained information that could not be interpreted as an object module.	Check for a typing error in the command line. Check the file type. The file being patched may not be an object file or may have been produced on a system other than RT-11.

BAD OVERLAY AT SEG # yyyyyy

LINKER	W	An overlay tried to store text outside its region; yyyyyy represents the segment number.	Check for an .ASECT in the overlay.

?BAD PPN?

FILEX	F	The DOS/BATCH user identification code was not in the form [nnn,nnn], where each nnn is an octal number less than or equal to 377 (octal).	Check the format of the user identification code.

?BAD RLD?

LINKER	F	There was an invalid relocation directory (RLD) command in the input file; the file is probably not a legal object module. The message occurs on pass 2 of the Linker.	Check for a typing error in the command line; verify that the correct files were specified as input. Reassemble or recompile to obtain a good object module and retry the operation. If the error persists, verify that the source code is correct.

BAD SEQUENCE ARGUMENT

BATCH	F	The identification number specified in a $SEQUENCE command was not numeric.	Reenter the command specifying the identification number as an unsigned decimal number.

?BAD SWITCH?

EXPAND	F	An unrecognized command string switch was specified.	No switches are used with EXPAND. Check for a typing error in the command line.
LINKER	F	The Linker did not recognize a switch specified in the command line.	If the bad switch occurred in the first command line, control returns to the CSI. Enter another command. If the bad switch occurred on a subsequent command line, the switch is ignored and processing continues. Reexamine the command line; check for a typing error.
MACRO	F	The switch specified was not recognized by the program.	Check for a typing error in the command line. Ensure that the switch is a valid listing control or function control (or CREF) switch.
PATCH	F	A switch other than /O or /M was typed.	Check for a typing error in the command line. Ensure that the switch entered is either /O or /M; no other switches are allowed.

BAD SWITCH

BATCH	F	The command line to the BATCH compiler contained an illegal switch.	Check for a typing error in the command line. Ensure that the switch indicated to the BATCH compiler is legal.

?BAD x SWITCH IGNORED?

LINKER	W	LINK did not recognize a switch (x) specified in the command line.	Check for a typing error in the command line. The switch is ignored and linking continues. Valid Linker switches are listed in Table 6-1 of the Linker chapter in the <u>RT-11 System Reference Manual</u> (DEC-11-ORUGA-C-D).

BAD VARIABLE

BATCH	F	The variable specified is not one of the characters A-Z.	Enter the variable as an alphabetic character.

BAD VID

BATCH	F	The volume identification specified in a $MOUNT command was not in the correct form.	Ensure that the equal sign and the name of the volume are included in the command.

BATCH FATAL ERROR

BATCH	F	A nonrecoverable error has occurred. This may indicate a problem in the software.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual. The system may have to be rebootstrapped. If the problem continues to occur, report the problem to Software Communications using an SPR (Software Performance Report). Include the contents of memory location 6012 (octal), along with a program listing and a machine readable source program, if possible.

BATCH HANDLER NOT RESIDENT

BATCH	F	The BATCH run-time handler was not loaded with the RT-11 LOAD command.	LOAD the BATCH handler before attempting to run BATCH.

BATCH STACK OVERFLOW

BATCH	F	There were too many nested $CALL commands in the BATCH stream.	Ensure that there are no more than 31 nested $CALL commands in the stream.

BC

BATCH	F	A Bad Code was found in the control file by the BATCH handler. This can happen when a .CTL file has been garbled or if an editing mistake was made by the programmer when altering or creating the file with EDIT.	Ensure that no editing errors have been introduced into the file. Recompile the .BAT file.

?BDR AT LINE xxxxx

BASIC	F	Item input from a DATA statement list by a READ statement was bad.	Ensure that the elements in the data list are in the correct format.

BExxxxxx

ODT	IN	A bad breakpoint entry occurred from location xxxxxx. This may indicate that a breakpoint routine was entered for no known breakpoint, that an illegal trace trap instruction occurred, that the T-bit was set in the status register, or that a jump to the middle of ODT occurred.	Reexecute the ODT commands, but in smaller steps, until the error in execution is reached.

?B-I/O ERROR

MBOOT MONITOR	F F	An I/O error occurred during system (or magtape) boot.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?B-NO BOOT ON VOLUME

MONITOR	F	No bootstrap was written on the volume.	Use the PIP /U switch to write the bootstrap.

?B-NO MONITOR.SYS

MONITOR	F	No monitor existed on the volume being booted.	The monitor filename must be MONITR.SYS. Refer to the RT-11 System Generation Manual (DEC-11-ORGMA-A-D) for instructions on renaming monitor files.

?B-NOT ENOUGH CORE			
MONITOR	F	There was not enough memory for the system being booted (e.g., attempting to boot F/B into 8K).	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
BLOCK xxxx IS BAD			
PIP	IN	The /K switch has been used to print on the terminal the absolute octal block numbers (xxxx) of the bad blocks.	Create a .BAD file to encompass the bad block. Refer to Chapter 4 of the RT-11 System Reference Manual (DEC-11-ORUGA-C-D) for detailed instructions.
?BOOT COPY?			
CBUILD MBUILD PIP	F F F	An error occurred during an attempt to write the bootstrap with the /U switch.	Retry the operation. If the error persists, use the PIP /K switch to check for bad blocks on the system device.
?BOTTOM ADDR WRONG?			
PATCH	F	The bottom address specified or contained in location 42 of an overlay file was incorrect.	Specify the correct address using the b;B command.
?BRT			
BASIC	A	The item entered to an input statement was bad.	Retype the item that caused the error and execution will continue.
?BSO AT LINE xxxxx			
BASIC	F	There was not enough room available in file buffers.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines;

			Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Reduce the size of arrays in memory to the size required (DIMension statement); Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

?BSW?

ASEMBL	F	The switch specified was not recognized by the program.	Check for a typing error in the command line. No switches are used with ASEMBL.

BUFFER STORAGE OVERFLOW AT LINE xxxxx

BASIC	F	There was not enough room available in file buffers.	Reduce the program size. This can be done by one of several procedures:

Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET;

Make maximum use of multiple statement lines;

Make efficient use of program loops, subroutines, and user-defined functions;

Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements;

Reduce the size of arrays in memory to the size required (DIMension statement);

Use virtual array files for arrays that are too large to fit into memory;

Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays;

Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use.

After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

BYTE RELOCATION ERROR AT xxxxxx

LINKER	W	The Linker attempted to relocate and link byte quantities, but failed. xxxxxx represents the address at	The relocated value is truncated to 8 bits and the Linker continues processing (for .SAV and .LDA files). For .REL

		which the error occurred. Failure is defined as the high byte of the relocated value (or the linked value) not being all zeroes.	files no truncation is performed and processing continues. Correct the source program so that there are no relocated byte quantities, reassemble, and relink.

******* C*

FORTRAN	IN	Illegal continuation. Comments cannot be continued and the first line of a program unit cannot be a continuation line.	The continuation line is ignored and compilation continues.

C

FORTRAN	F	Constant subscript overflow. Too many subscripts were employed in a statement.	Simplify the statement.

** CB ALMOST FULL **

EDIT	A	The command currently being entered is within 10 characters of exceeding the space available in the Command Buffer.	Complete the command using less than 10 characters if possible. Otherwise, type ALTMODE twice to execute that portion of the command line already completed and then enter the remainder as a second command.

?CB FULL?

EDIT	IN	The command exceeded the space allowed for a command string in the Command Buffer.	Any additional characters typed are ignored. Type ALTMODE twice to execute that portion of the command line already completed, or type CTRL X to abort the command.

?C-CHAIN-ONLY-CUSP?

CREF	F	An attempt was made to either "R CREF" or to "START" a copy of CREF that was in memory. CREF can only be chained to.	Use the appropriate language processor to invoke CREF.

?C-CRF FILE ERROR?			
CREF	F	An input error occurred while accessing "DK:CREF.TMP", the temporary file passed to CREF.	Rerun the language processor to create a good CREF input file.
?C-DEVICE?			
CREF	F	An invalid device was specified to CREF.	This represents a system error. Before trying to output a CREF listing to magtape or cassette, make sure the handler is loaded first. Try to reproduce the error and report it to Software Communications using an SPR (Software Performance Report); include a program listing and a machine readable source program, if possible.
?C-LST FILE ERROR?			
CREF	F	An output error occurred while attempting to write the cross-reference table to the listing file. This may indicate that not enough room is available for the listing file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual. Refer to Section 3.0, for information on how to increase memory space.
?CHK SUM?			
PIP	IN	A checksum error occurred during a formatted binary transfer.	Check for a typing error in the command line. Ensure that the correct file is being transferred. The error may also indicate that data has been lost from the input file. Retry the operation; if the error persists, obtain a new copy of the .OBJ or .LDA file.
COMMAND NOT UNIQUE			
BATCH	F	$JOB/UNIQUE has been specified and the spelling of a command was not the shortest unique spelling.	Ensure that the command includes enough characters to make it unique.

COMMON BLOCK EXCEEDS MAXIMUM SIZE

FORTRAN	IN	An attempt was made to allocate more space to COMMON than is physically addressable (greater than 32K words).	Correct the statement in error.

CONSTANT IN FORMAT STATEMENT NOT IN RANGE

FORTRAN	IN	An integer constant in a FORMAT statement was not in the proper range.	Check that all integer constants are within the legal range (1 to 255).

?COR OVR?

CBUILD FILEX	F F	There was insufficient main storage for buffers and input list expansion.	In each case, refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Try copying the files one at a time, without using the "wild-card" (*.*) construction on input.
MBUILD PIP	F F	Memory overflow occurred resulting from too many device and/or file specifications (usually in "wild-card" operations) and no room for buffers.	
SRCCOM	F	There was not enough memory to hold a particular difference section.	

?CORE?

ASEMBL	F	There were too many symbols in the program being assembled.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
LINKER	F	There was not enough memory to accommodate the command, the symbol table or the resultant load module.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.

?CSECT ERROR?

LIBR	F	The user has extended beyond the .CSECT space allowed for an object	Use /G to eliminate extra .CSECT names from the directory.

		module to be placed in the library (i.e., the object module contains more than 127(decimal) .CSECTs).	

CTn: PUSH REWIND OR MOUNT NEW VOLUME

PIP	A	During an I/O operation, the end of the cassette currently mounted on drive n has been reached.	Mount a new cassette (or the next cassette in sequence) on drive n and the operation will continue automatically.

******* D*

MACRO	IN	Multiply-defined label referenced. Reference was made to a label (not a local label) that was defined more than once.	Correct the source progam so that each label is defined only once, and reassemble.

D

MACRO	IN	Multiply-defined label referenced. Reference was made to a label (not a local label) that was defined more than once.	Correct the source progam so that each label is defined only once, and reassemble.

DANGLING OPERATOR

FORTRAN	IN	An operator (+, -, *, /, etc.) was missing an operand. (For example, (I=J+).)	Correct the statement in error.

?DAT?

MONITOR	F	The DATE command argument was illegal, or the date was requested when it had not yet been set.	Check for a typing error in the command line. Enter the date using the correct format (DAT dd-mm-yy).

?DCE AT LINE xxxxx

BASIC	F	The device channel number specified for a sequential or virtual memory file was out of range (1-7), or has been opened, or an OPEN statement	Check the channel number specified in the OPEN statement; check that the device is an RT-11 directory-structured device.

		tried to open a virtual memory file on a nonRT-11 directory-structured device.	

DEFECTIVE DOTTED KEYWORD

FORTRAN	IN	A dotted relational operator was not recognized, or possible misuse of a decimal point occurred.	Check the format of relational operators; correct the statement in error.

?DEV FUL?

CSI	F	This message is caused by a user program utilizing .CSIGEN and indicates that the output file did not fit on the device specified.	In each case, refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.
MBUILD PIP	F F	There was no room on the device for the file.	

?DEV FULL?

CBUILD FILEX	F F	There was no room in the directory for the filename or there was no room on the output device for the file. The filename is not placed in the output device directory.	In each case, refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.
LIBR	F	The device was full; LIBR was unable to create or update the indicated library file.	

DEVICE CHANNEL ERROR AT LINE xxxxx

BASIC	F	The device channel number specified for a sequential or virtual memory file was out of range (1-7), or has been opened, or an OPEN statement tried to open a virtual memory file on a nonRT-11 directory-structured device.	Check the channel number specified in the OPEN statement; check that the device is an RT-11 directory-structured device.

DEVICE NOT READY AT LINE xxxxx			
BASIC	F	An I/O device referenced by an OLD, SAVE, or PRINT command was not on-line, or the file did not contain any legal BASIC program lines.	Ensure that the device is on-line. Check the format of the statements within the program being accessed. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
?DIR ERR?			
FILEX	F	An error occurred while reading or looking up the directory of the input device, or the input device did not have the proper file structure.	Check for a typing error in the command line. Verify that the input device has the correct structure. If so, a hard error condition exists. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
*?*DIR FULL*?*			
EDIT	F	No room existed in the device directory for the output filename. (This message occurs following an EB or EW command.)	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.
DISMOUNT ERROR			
BATCH	F	The logical device name specified did not exist.	Ensure that the device has been assigned with a $MOUNT command.
DIVISION BY Ø AT LINE xxxxx			
BASIC	F	The program attempted to divide some quantity by 0.	Correct the program logic.
?DNR AT LINE xxxxx			
BASIC	F	An I/O device referenced by an OLD, SAVE, or PRINT command was not on-line, or the file did not contain any legal BASIC program lines.	Ensure that the device is on-line. Check the format of the statements within the program being accessed. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

*DO TERMINATOR *** PRECEDES DO STATEMENT*			
FORTRAN	IN	The statement specified as the terminator of a DO loop did not appear after the DO statement. (*** represents the label on the DO terminator statement.)	Correct the program logic.
?DUMP ERROR?			
PATCHO	F	An input/output error occurred while dumping a module. This indicates a bad type code in a formatted binary block.	Check for a typing error in the command line. Check the object module; it may have been produced by a system other than RT-11.
?DVØ AT LINE xxxxx			
BASIC	F	The program attempted to divide some quantity by 0.	Correct the program logic.
******* E*			
FORTRAN	IN	Missing END statement.	A .END statement is supplied by the Compiler if an end-of-file is encountered. Correct the source program.
MACRO	IN	END directive not found.	An END directive is generated. Correct the source program so that it contains an END directive.
E			
MACRO	IN	END directive not found.	An END directive is generated. Correct the source program so that it contains an END directive.
END BATCH			
BATCH	IN	A BATCH job has been terminated.	Control returns to the monitor.

ENTER CHECKSUM: xxxxxx

PATCHO A

This message is printed in response to the PATCHO EXIT command.

Enter the correct checksum as a 6-digit octal number. If the checksum does not match that computed by PATCHO, the PAUSE--?BAD PATCH? error message is printed.

ENTRY POINT:
ent1 <CR>
ent2 <CR>
<CR>

LIBR A

This message is printed following use of the Librarian /G switch. (entl, ent2, etc. represent the names of entry points.)

Enter the names of the entry points entl, ent2, etc.) to be deleted from the library, following each with a carriage return. A single carriage return terminates the list.

*?*EOF*?*

EDIT F

A Read, Next or file search command was attempted and no input data was available.

The end of the file has been reached. Close the file using EX or EF; reopen it if more editing remains to be done.

EOF WITH NO EOJ

BATCH F

A file was not terminated with a $EOJ command.

Insert a $EOJ command as the last statement in the BATCH job.

?^ ER AT LINE xxxxx

BASIC F

The program tried to compute the value A^B, where A<0 and B was not an integer or B>256. This produces a complex number which is not representable in BASIC. This message may also occur if the argument to the EXP function was greater than 87.

Correct the program logic.

?ER RD DIR?

CBUILD	F	A hardware error was reported while reading the directory.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
MBUILD	F		
PIP	F		

?ER RD OVLY?

MONITOR	F	A hardware error occurred while reading a KMON overlay to process the current command; this indicates that the system file MONITR.SYS is unreadable.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?ER WR DIR?

MBUILD	F	A hardware error was reported while writing the directory.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
PIP	F		

?"<>"ERR?

EDIT	F	Iteration brackets were nested too deeply or used illegally, or the brackets were not matched.	Ensure that all brackets are properly matched and that the number of nested brackets does not exceed 20 levels.

?ERR nn [text]

FORTRAN		Error diagnostics detected by the Object Time System are reported in either short form (where nn is a decimal error condition), or long form (where "text" is an added short error description). The user may optionally change both the format of a message and its type. Refer to the RT-11/RSTS/E FORTRAN IV User's Guide (DEC-11-LRRUA-A-D) for instructions.	These errors are listed in the beginning of the system message section according to their decimal value. Refer in each case to the decimal portion of the message for a description of the error condition.

? *** ERROR *** text			
EXPAND	W	Indicates nonfatal errors in the input file; "text" is a short explanation of the error.	Refer to the individual "text" listing.
^ ERROR AT LINE xxxxx			
BASIC	F	The program tried to compute the value A^B, where A<0 and B was not an integer or B>256. This produces a complex number which is not representable in BASIC. This message may also occur if the argument to the EXP function was greater than 87.	Correct the program logic.
?ERROR ERROR?			
LINKER	F	An internal error occurred while the Linker was in the process of recovering from a previous system user error.	Retry the operations that produced this error; if it recurs, report the error to Software Communications using an SPR (Software Performance Report); include a program listing and a machine-readable source program, if possible.
?ERROR IN FETCH?			
LINKER	F	The device was not available.	Verify that the device is valid for the system in use.
[name] ERRORS: nnn, WARNINGS: mmm			
FORTRAN	IN	For each program unit, name represents .MAIN. for main program units, or the function or subroutine name for subprogram units. nnn totals the number of errors and mmm the number of warnings.	Correct all initial or secondary phase error conditions. Warnings frequently indicate a programming error on the part of the user.

?ETC AT LINE xxxxx

BASIC	F	The expression being evaluated caused the stack to overflow. This may occur because parentheses were nested too deeply or because of many complex user-defined functions.	The degree of complexity that produces this error varies according to the amount of space available in the stack at the time. Breaking the statement into several simpler ones eliminates the error. Consult the BASIC/RT-11 System Release Notes (DEC-11-LBRNA-A-D) for a method of patching BASIC to increase the stack size.

*EXPECTING LEFT PARENTHESES AFTER *****

FORTRAN	IN	An array name or function name reference was not followed by a left parenthesis. (**** represents the array or function name.)	Correct the statement so that the left parentheses is included.

EXPRESSION TOO COMPLEX AT LINE xxxxx

BASIC	F	The expression being evaluated caused the stack to overflow. This may occur because parentheses were nested too deeply or because of many complex user-defined functions.	The degree of complexity that produces this error varies according to the amount of space available in the stack at the time. Breaking the statement into several simpler ones eliminates the error. Consult the BASIC/RT-11 System Release Notes (DEC-11-LBRNA-A-D) for a method of patching BASIC to increase the stack size.

?EXT NEG?

PIP	F	A /T command specified that the file should be made smaller than its current size.	The operation is ignored. Enter another command.

EXTRA CHARACTERS AT END OF STATEMENT

FORTRAN	IN	All the necessary information for a syntactically correct FORTRAN statement was found on the line, but more information exists.	Check that a comma is not missing from the line and that no unintentional continuation signal occurs on the next line.

F?			
MONITOR	W	A CTRL F was typed under the F/B Monitor and no foreground job exists.	Control remains with the background job. Type another command.
?F ACTIVE?			
MONITOR	F	An attempt was made to execute an FRUN or UNLOAD command when a foreground job already existed and was active.	Wait for the foreground job to finish before unloading handlers and starting a new foreground job.
FATAL ERROR n			
FORTRAN	F	Diagnostics in this format report hardware error conditions and conditions that may require rewriting the program. "n" is a single letter representing an error code.	Refer to the individual listing of "n".
?FDE AT LINE xxxxx			
BASIC	F	An attempt was made to write an element on an integer virtual memory file outside the range -32768 to +32768.	Ensure that all integer virtual file assignments are within the legal range.
FE			
BATCH	F	A forced end occurred due to the appearance in the .CTL file of an illegal \F followed by a carriage return, or BATCH was terminated from the console by a \F followed by a carriage return.	Insert another BATCH control directive after the \F to prevent forced termination.
?FEATURE NOT IMP?			
FILEX	F	An operation was attempted which FILEX cannot perform (e.g., zeroing an RT-11 device).	The operation is ignored. Check for a typing error in the command line. Enter another command.

?FG ACTIVE?			
FILEX	F	An attempt was made to use /T when a foreground job was active.	The transfer is not allowed until the foreground job is terminated and unloaded via UNLOAD FG.
?FG PRESENT?			
PIP	F	An attempt was made to use /O or /S while a foreground job was still in memory.	The operation is not allowed until the foreground job is terminated and unloaded via UNLOAD FG.
?FIL NAM?			
FILEX	F	The output filename was invalid or null.	Check for a typing error in the command line. Verify that an output filename was specified in the correct format and that it contains no illegal characters.
?FIL NOT FND?			
CBUILD CSI	F F	The input file was not found.	In each case, check for a typing error in the command line; verify that the filename exists as entered in the command line, and retry the operation.
FILEX	F	The input file was not found, or the "wild-card" (*.*) construction matched none of the existing files.	
LIBR	F	One of the input files indicated in the command line was not found.	
MONITOR	F	The file specified in an R, RUN, GET, or FRUN command was not found.	
MBUILD PIP	F F	The input file specified was not found during a delete, copy, or rename operation, or no input files with the expected name or extension were found during a *.* expansion.	

?FILE?			
MONITOR	F	No file was named where one was expected (for example, RUN<CR>).	Check for a typing error in the command line; check the format of the monitor command and retry the operation inserting the proper filename.
FILE DATA ERROR AT LINE xxxxx			
BASIC	F	An attempt was made to write an element on an integer virtual memory file outside the range -32768 to +32768.	Ensure that all integer virtual file assignments are within the legal range.
*?*FILE FULL*?*			
EDIT	F	Available space for the output file was full.	If the ER and EW commands have been used to open the I/O files, type a CTRL C and a CLOSE monitor command to save the data already written. If the EB command has been used, type CTRL C (do not type CLOSE). All edits must be reentered.
FILE I/O ERROR AT LINE xxxxx			
BASIC	F	A hard error occurred during an I/O operation.	All files are automatically closed. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
?FILE NOT FND?			
LINKER	F	The input file was not found.	Check for a typing error in the command line. Verify that the filename exists as entered in the command line and retry the operation.

*?*FILE NOT FND*?*

EDIT	F	An attempt was made to open a nonexistent file for editing.	Check for a typing error in the command line. Verify that the file name exists as entered in the command line and retry the operation.

?FILE NOT FOUND?

MSBOOT	F	The file specified was not found on the magtape.	Check for typing error in the command line. Verify that the filename exists as entered in the command line and retry the operation.

FILE NOT FOUND AT LINE xxxxx

BASIC	F	The file requested was not found on the specified device.	Check for a typing error in the command line. Verify that the device, filename, and extension exist as specified.

FILE NOT OPEN AT LINE xxxxx

BASIC	F	The sequential or virtual memory file referenced was not open.	Open the file before attempting to access it.

FILE TOO SHORT AT LINE xxxxx

BASIC	F	The sequential file space allocated to an output file was inadequate.	Use the PIP /S or /D switch to ensure sufficient file space.

%FILES ARE DIFFERENT

SRCCOM	IN	This message indicates that the files compared were different.	Verify that the files were expected to be different; otherwise the wrong files may have been compared.

?FIO AT LINE xxxxx

BASIC	F	A hard error occurred during an I/O operation.	All files are automatically closed. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

FLOATING CONSTANT NOT IN RANGE

FORTRAN	F	A floating constant in an expression was too close to zero to be represented in the internal format.	Use zero if possible.

?FNF AT LINE xxxxx

BASIC	F	The file requested was not found on the specified device.	Check for a typing error in the command line. Verify that the device, filename, and extension exist as specified.

?FNO AT LINE xxxxx

BASIC	F	The sequential or virtual memory file referenced was not open.	Open the file before attempting to access it.

FOR WITHOUT NEXT AT LINE xxxxx

BASIC	F	The program contained a FOR statement without a corresponding NEXT statement to terminate the loop.	Insert NEXT statements to correspond to each FOR statement within the program.

?FORLIB NOT FND?

LINKER	F	An attempt was made to link the FORTRAN library with other object modules in the command line, but FORLIB.OBJ was not found.	Check for a typing error in the command line. Verify that FORLIB.OBJ exists on the system device, and that the system device is running on unit 0.

?FTS AT LINE xxxxx

BASIC	F	The sequential file space allocated to an output file was inadequate.	Use the PIP /S or /D switch to ensure sufficient file space, or specify the number of blocks to be allocated using the n construction on the OPEN FOR INPUT statement.

?FWN AT LINE xxxxx

BASIC	F	The program contained a FOR statement without a corresponding NEXT statement to terminate the loop.	Insert NEXT statements to correspond to each FOR statement within the program.

?GND AT LINE xxxxx

BASIC	F	Program GOSUBS were nested to more than 20 levels.	Limit GOSUB nesting to 20 levels.

GOSUBS NESTED TOO DEEPLY AT LINE xxxxx.

BASIC	F	Program GOSUBS were nested to more than 20 levels.	Limit GOSUB nesting to 20 levels.

******* H*

FORTRAN	IN	A Hollerith string or quoted literal string was longer than the remainder of the statement.	The statement is ignored and compilation continues. Correct the statement in error and recompile.

?HARD I/O ERROR?

LINKER MSBOOT MTINIT	F F F	A hardware error was reported during an I/O operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

*?*HDW ERR*?*

EDIT	F	A hardware error was reported during an I/O operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

******* I*

FORTRAN	IN	A non-FORTRAN character was used. The line contained a character that was not in the FORTRAN character set and was not in a Hollerith string or comment line.	The character is ignored and compilation continues. Correct the line in error and recompile.

MACRO	IN	Illegal character detected.	Illegal characters that are also nonprinting are replaced by a ? on the listing and ignored. Correct the source program and reassemble.
I			
MACRO	IN	Illegal character detected.	Illegal characters that are also nonprinting are replaced by a ? on the listing and ignored. Correct the source program and reassemble.
?IDF AT LINE xxxxx			
BASIC	F	The define function statement contained an error.	Correct the syntax of the DEF statement.
?IDM AT LINE xxxxx			
BASIC	F	A syntax error occurred in a dimension statement.	Correct the syntax of the dimension statement. An array can only be dimensioned once and can have at most two dimensions.
?ILL ARG?			
EDIT	F	The argument specified was illegal for the command used; a negative argument was specified where a positive one was expected; the argument exceeded the range + or - 16,383.	Check the command format for the proper argument usage and reenter the command correctly.
?ILL ASECT?			
LINKER	F	An attempt was made to place an .ASECT above 1000 in a foreground link or to place an .ASECT into an overlay foreground link.	Correct the source program so that the restriction on .ASECTs is observed.

?ILL CMD?

CSI	F	There was a syntax error in the command line.	Check for a typing error in the command line. Check the format of the monitor command and reenter it.
EDIT	F	EDIT did not recognize the command line specified, or ED was not the first command used to activate the display hardware.	If the display hardware is to be used, recall the Editor and type ED before entering any other editing commands. Check for a typing error in the command line. Check the format of any editing command that produces this error, and reenter the command correctly.
FILEX		The command entered was illegal for one of the following reasons: the length of the command line exceeded 72 characters; the command line was not in the proper CSI format; the UIC exceeded the allowed number of characters, or [was used without]; a "wild-card" (*.*) construction was used on a sequential-access device; no output or no input file was specified for a copy operation; more than one filename construction (dev:filnam.ext) was specified on either side of the = or < sign.	Check for a typing error in the command line. Verify that the format of the command line is correct and that the UIC is in the proper format, and retry the operation.
LIBR	F	An illegal command was used in the command line.	Check for a typing error in the command line. Check the format of the command line and retry the operation.
MONITOR	F	An illegal Keyboard Monitor command was used or the command line was too long.	Check for a typing error in the command line. Check the format of the command line and verify that it does not exceed 72 characters. Retry the operation.

MBUILD PIP	F F	The command line specified was not syntactically correct for one of the following reasons: a required device name was missing; a switch argument was too large, a filename was specified where inappropriate; a nonfile-structured device was specified for a file-structured operation.	Check for a typing error in the command line. Verify that the format and syntax of the command line are correct and retry the operation.

xxxxxx ILL DEL

LIBR	W	An attempt was made to delete from the library's directory a module or an entry point that does not exist; xxxxxx represents the module or entry point name.	Check for a typing error in the command line. The entry point name or module name is ignored and processing continues.

?ILL DEV?

CBUILD	F	An illegal cassette unit number was specified, or cassette operations were requested for both input and output.	Check for a typing error in the command line. Reenter the command correctly.
CSI	F	The device specified does not exist.	Check for a typing error in the command line. Ensure that the device indicated is a valid device.
FILEX	F	The device handler was not found, an invalid or illegal device name (e.g., MT or CT) was used, or one of the following was attempted: RK or DT was not used for DOS/BATCH (RSTS) input in a copy operation; DT was not used for DOS/BATCH (RSTS) output in a zero or delete operation; DT was not used for DOS/BATCH (RSTS) output in a copy operation; DT was not used for DECsystem-10 input in any operation.	Check for a typing error in the command line. Check that nesessary device handlers are present on the system, and that the device indicated is a legal device name and valid for the operation indicated.

LIBR	F	An illegal device was specified in the command line.	Check for a typing error in the command line. Verify that the device indicated is valid and reenter the command.
MBUILD MONITOR PIP	F F F	An illegal or nonexistent device was indicated, an operation illegal for the specified device was attempted, or an attempt was made to make a device handler resident for use with a foreground job (dev=F) when the Single-Job Monitor was running.	Check for a typing error in the command line. Verify that the device indicated is valid. Note that devices for R, RUN, GET, SAVE, FRUN must be random-access devices. (If the device handler size is too large, an ?M-BAD FETCH error message is printed.) The dev=F (and dev=B) construction is valid only under the F/B Monitor. Reenter the command.
*?*ILL DEV*?*			
EDIT	F	An attempt was made to open a file on an illegal device, or to use display hardware when none was available.	Check for a typing error in the command line. Verify that the device indicated is valid, or that display hardware exists and is not already in use by the other job.
?ILL DIR?			
PIP	F	The device did not contain a properly initialized directory structure (end-of-tape file on cassette; empty file directory on other devices).	Initialize the device with /Z before using it the first time.
xxxxxx ILL INS			
LIBR	W	An attempt was made to insert into a library a module that contains the same entry point as an existing module. xxxxxx represents the entry point name.	The entry point is ignored, but the module is still inserted into the library. No user action is necessary.

?ILL MAC?

EDIT	F	Delimiters in an M command were improperly used, or an attempt was made to enter an M command during execution of a Macro or an EM command while an EM was already in progress.	Check for a typing error in the command line. Ensure that the character used for the delimeters does not appear in the MACRO itself. Do not attempt to enter M or EM until the current macro has finished executing.

?*ILL NAME*?

EDIT	F	The filename specified in an EB, EW, or ER command was illegal.	Check for a typing error in the command line. Ensure that the dev:filnam.ext[n] specification does not exceed 19 characters and is in the proper format. Verify that an input filename exists as entered.

?ILL REN?

MBUILD PIP	F F	An illegal rename operation was attempted.	Check for a typing error in the command line. Verify that the same device name is specified on both the input and output sides of the command string.

xxxxxx ILL REPL			
LIBR	W	An attempt was made to replace in the library file a module that does not already exist. xxxxxx represents the module name.	The module is ignored and the library is built without it.
?ILL SWT?			
FILEX	F	An illegal switch was used in a command line.	Check for a typing error in the command line. Use only those switches listed in Table J-1 of the <u>RT-11 System Reference Manual</u> (DEC-11-ORUGA-C-D); all others are illegal for FILEX.
MBUILD	F	An illegal switch or switch combination was used in a command line.	Check for a typing error in the command line. Use only those switches listed in Chapter 4, Section 4.3 of the <u>RT-11 System Generation Manual</u> (DEC-11-ORGMA-A-D).
PIP	F	An illegal switch or switch combination was used in a command line.	Check for a typing error in the command line. Use only those switches listed in Table 4-1 of the <u>RT-11 System Reference Manual</u> (DEC-11-ORUGA-C-D). Any number of switches may be used in a command line as long as only one operation switch (insertion, deletion, etc.) is represented.
ILLEGAL '+'			
BATCH	F	The + construction was used when not allowed (e.g., in a $RUN or $BASIC input file descriptor), or there was a + in an output file descriptor, or a + terminated a file descriptor.	Verify that the + is used either to indicate a positive value in a switch, or to separate multiple file descriptors.

ILLEGAL ADJACENT OPERATOR

FORTRAN		Two operators (*, /, logical operators, etc.) were illegally placed next to each other.	Correct the statement in error.

ILLEGAL CHARACTER

BATCH	F	The character specified was not used in proper context.	Check the log file to determine the character in error.

?ILLEGAL COMMAND?

MTINIT	F	A command other than that expected was typed.	Check for a typing error in the command line. Enter only the command: MTn:=<CR>
PATCHO	F	A command line was not recognized, or it was not in the proper format for the particular operation.	Check for a typing error in the command line. Ensure that the format of the command line is correct for the specified operation.

ILLEGAL COMMAND LINE

BATCH	F	The command line to the BATCH compiler was incorrect.	Check for a typing error in the command line. Check the format for the command in use, and reenter the command.

ILLEGAL DEF AT LINE xxxxx

BASIC	F	The define function statement contained an error.	Correct the syntax of the DEF statement.

?ILLEGAL DEVICE?

MTINIT	F	The device specified was not a TM11 or TJU16 magtape unit.	Check for a typing error in the command line. Reenter the command, specifying a legal magtape unit.

ILLEGAL DEVICE

BATCH	F	An illegal device or a non-existent device was specified in the command line.	Check for a typing error in the command line. Verify that the device exists as entered in the command line, or use another device, and reenter the command.

ILLEGAL DIM AT LINE xxxxx

BASIC	F	A syntax error occurred in a dimension statement.	Correct the syntax of the dimension statement. An array can only be dimensioned once and can have at most two dimensions.

*ILLEGAL DO TERMINATOR ORDERING AT LABEL *****

FORTRAN	IN	DO loops were incorrectly nested. (**** represents the label name.)	Verify that the range of each DO loop lies completely within the range of the next outer loop.

*ILLEGAL DO TERMINATOR STATEMENT *****

FORTRAN	IN	A DO terminator statement was not valid. (**** represents the label of the DO terminator statement.)	Verify that the DO terminator statement is not a GO TO, arithmetic IF, RETURN, or another DO statement, or a logical IF containing one of these statements.

ILLEGAL ELEMENT IN I/O LIST

FORTRAN	IN	An item, expression, or implied DO specifier in an I/O list was not in the correct syntax.	Correct the I/O list.

?ILLEGAL FILENAME?

MSBOOT	F	The name entered was not a legal RT-11 filename.	Check for a typing error in the comand line. The file must be in .SAV format and reside on magtape. Reenter the command using a valid file.

ILLEGAL LOG DEVICE

BATCH	F	Magtape, cassette, or a read-only device (e.g., PR:) was specified as the log device.	Check for a typing error in the command line. Assign the log device to a suitable device.

ILLEGAL NOW

BASIC	F	An attempt was made to execute an INPUT statement while in immediate mode.	Assign a line number to the INPUT statement and use an immediate mode GOTO to execute it, or use an immediate mode assignment statement.

ILLEGAL READ AT LINE xxxxx

BASIC	F	An attempt was made to read on a sequential file open for output.	Ensure that a file is open for input.

ILLEGAL STATEMENT IN BLOCK DATA

FORTRAN	IN	An illegal statement was found in a BLOCK DATA subprogram.	Verify that no FORMAT or executable statements occur in a BLOCK DATA subprogram.

ILLEGAL STATEMENT ON LOGICAL IF

FORTRAN	IN	The statement contained in a logical IF was not valid.	Verify that the statement is not a DO statement or another logical IF.

ILLEGAL SWITCH

BATCH	F	The switch name specified was not a legal RT-11 BATCH switch or was not legal for this field.	Use only valid switch abbreviations for the field to which they apply.

ILLEGAL SWITCH COMBINATION

BATCH	F	More than one switch of the same type existed on the same command line.	Ensure that only one switch from the following combinations is used: /MACRO, /INPUT, /SOURCE; /FORTRAN, /INPUT, /SOURCE; /BASIC, /INPUT, /SOURCE.

ILLEGAL TYPE FOR OPERATOR

FORTRAN	IN	An illegal variable type was used with an exponentiation or logical operator.	Check that the variable type is valid for the operation in question.

ILLEGAL USAGE OF OR MISSING LEFT PARENTHESIS

FORTRAN	IN	A left parenthesis was required but not found, or a variable reference or constant was illegally followed by a left parenthesis.	Correct the format of the statement in error.

?ILN

BASIC	F	An attempt was made to execute an INPUT statement while in immediate mode.	Assign a line number to the INPUT statement and use an immediate mode GOTO to execute it, or use an immediate mode assignment statement.

?ILR AT LINE xxxxx

BASIC	F	An attempt was made to read on a sequential file open for output.	Ensure that a file is open for input.

?IN ER?

MBUILD	F	A hardware error occurred while reading the file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
PIP	F		

?IN ERR?

CBUILD	F	A hardware error was reported during an input operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual. (Under DUMP, the /G switch may be used to ignore input errors.)
DUMP	F		
FILEX	F		
LIBR	F		
SRCCOM	F		

IN LINE nnnnn MSG# m text

FORTRAN		These are errors reported by the secondary phase of the compiler. nnnnn is the internal sequence number of the statement, m specifies the error number, text is an explanation of the error.	See the individual listings of 'text' for a description of each message.

?INCORRECT FILE SPEC?

PATCH	F	The response to the "FILE NAME--" message was not in the correct format.	Check for a typing error in the command line. Ensure that the correct device, filename and switch option are used, and retry the operation.

?INPUT ERROR?

EXPAND	F	A hardware error was reported during an input operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

INPUT ERROR

BATCH	F	A hardware error was reported while attempting to read the compiler input file (.BAT).	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

INPUT FILE?

BATCH	F	An input file descriptor was not specified to the BATCH compiler command line.	Check for a typing error in the command line. Specify an input file in the command line.

?INSUFFICIENT CORE?

EXPAND	F	There was not enough memory to store macro definitions.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
MACRO	F	There were too many symbols in the program being assembled.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
PATCH	F	PATCH did not have enough memory to hold the file's device handler plus the internal "segment table".	Try patching the file on the system device so that handlers need not be loaded. Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.

INTEGER OVERFLOW

FORTRAN	IN	An integer constant or expression was outside the legal range.	Correct the value of the integer constant or expression so that it falls within the legal range (-32767 to +32767).

INVALID COMPLEX CONSTANT

FORTRAN	IN	A complex constant has been improperly formed.	Correct the statement in error.

*INVALID DIMENSIONS FOR ARRAY *****

FORTRAN	IN	An attempt was made while dimensioning an array to explicitly specify zero as one of the dimensions. (**** represents the array name.)	Verify that zero is not used as a dimension.

INVALID EQUIVALENCE

FORTRAN	IN	An illegal equivalence, or an equivalence that was contradictory to a previous equivalence, was encountered.	Correct the program logic.

INVALID FORMAT SPECIFIER

FORTRAN	IN	A format specifier was illegally used.	Correct the statement so that the format specifier is the label of a FORMAT statement or an array name.

INVALID IMPLICIT RANGE SPECIFIER

FORTRAN	IN	An illegal implicit range specifier was encountered.	Verify that the implicit range specifier indicates alphabetic characters in alphabetic order.

INVALID LOGICAL UNIT

FORTRAN	IN	A logical unit reference was incorrect.	Correct the logical unit reference so that it is an integer variable or constant within the legal range (1 to 99).

INVALID OCTAL CONSTANT

FORTRAN	IN	An octal constant was too large or contained a digit other than 0-7.	Correct the constant so that it contains only legal digits and falls within the range 0 to 177777.

INVALID OPTIONAL LENGTH SPECIFIER

FORTRAN	IN	A data type declaration optional length specifier was illegal (for example, REAL*4 and REAL*8 are legal, but REAL*6 is not).	Correct the statement so that it contains only a valid data type declaration length.

INVALID RADIX5Ø CONSTANT

FORTRAN	IN	An illegal character was detected in a Radix-50 constant.	Verify that only characters from the Radix-50 character set are used in a Radix-50 constant.

INVALID RECORD FORMAT

FORTRAN	IN	The third parenthetical argument in a DEFINE FILE statement must be a single character U.	Correct the DEFINE FILE statement.

?INVALID RELOC REG?

PATCH	F	An attempt was made to reference a relocation register outside the range 0-7.	Check for a typing error in the command line. Relocation registers must be set within the range 0-7.

?INVALID SEG NO?

PATCH	F	The segment number S: did not exist.	Check for a typing error in the command line. Recheck the Linker load map and command string to determine the overlay structure.

INVALID STATEMENT LABEL REFERENCE

FORTRAN	IN	Reference was made to a statement number that was of illegal construction (for example, GO TO 999999 is illegal since the statement number is too long).	Check that the statement number consists of from 1 to 5 decimal digits placed in the first five columns of a statement's initial line, and does not contain only zeroes.

INVALID SUBROUTINE OR FUNCTION NAME

FORTRAN	IN	A name used in a CALL statement or function reference was not valid (for example, use of an array name in a CALL statement subroutine name reference).	Verify that the name specified in the statement is spelled correctly.

INVALID TARGET FOR ASSIGNMENT

FORTRAN	IN	The left side of an arithmetic assignment statement was not a variable name or array element reference.	Correct the statement in error.

INVALID TYPE SPECIFIER

FORTRAN	IN	An unrecognizable data type was used.	Verify that the data type indicated is valid.

INVALID USAGE OF FUNCTION OR SUBROUTINE NAME

FORTRAN	IN	A function name appeared in a DIMENSION, COMMON, DATA, EQUIVALENCE, or Data Type Declaration statement.	Correct the statement in error.

INVALID VARIABLE NAME

FORTRAN	IN	A variable name was missing, contained an illegal character, or did not begin with an alphabetic character.	Correct the statement in error.

?I/O ERROR ON CHANNEL n?

MACRO	F	A hardware error occurred while attempting to read from or write to the device on the channel specified in the message. (Channel numbers are assigned to files in the manner described in Section 9.4.7 of the <u>RT-11 System Reference Manual</u>, (DEC-11-ORUGA-C-D.)	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

IO

BATCH	F	An input or output error occurred when the BATCH handler was attempting to read the .CTL file or write to the log file. The probable cause of this error is a log file overflow.	Rerun the BATCH stream, specifying a larger log file with the square bracket construction (the default log size is 64 decimal blocks).

****** *K*

FORTRAN	IN	Illegal statement label definition. An illegal (non-numeric) character was found in statement label.	The illegal statement label is ignored and compilation continues. Correct the program and recompile.

****** *L*

FORTRAN	IN	Line too long. There were more than 80 characters in a line (this diagnostic is issued before the line containing too many characters).	The line is truncated to 80 characters and compilation continues.
MACRO	IN	Line buffer overflow; the input line was greater than 132 characters.	Extra characters on a line are ignored in terminal mode. Correct the source program.

L

MACRO	IN	Line buffer overflow; the input line was greater than 132 characters.	Extra characters on a line are ignored in terminal mode. Correct the source program.

LABEL ON DECLARATIVE STATEMENT

FORTRAN	IN	A label was indicated on a declarative statement.	Correct the program so that declarative statements do not have labels.

?LDA FILE ERROR?

LINKER	F	The device was full or a hardware error was reported on the device specified for .LDA output.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Check the procedures for hard error conditions listed in Section 1.0.

?LIBR FIL ILL REPL?

LIBR	F	The command line specified that a library file be replaced by another library file.	Check for a typing error in the command line. Only object modules can be replaced in a library file. Enter another command.

LIBRARY SEARCH:

sym1 <CR>
sym2 <CR>
<CR>

LINKER	A	This message is printed following use of the Linker /I switch. (syml,sym2, etc. represent global symbols to be entered at the keyboard.)	Specify the global symbols to be included in the load module. Type a carriage return to enter each symbol; terminate the list by typing a single carriage return.

LINE TOO LONG

BASIC	F	The line typed was longer than 132 characters.	The line buffer overflows. Break the line into two or more lines, or if reading from a file, ensure that the file contains only legal BASIC program lines.
BATCH	F	The input line entered is greater than 80 characters.	Correct the input line.
EXPAND	IN	A line has become longer than 132 characters.	The line is truncated to 132 characters.

LINE TOO LONG TO TRANSLATE AT LINE xxxxx

BASIC	F	The line just entered exceeded the area available for translation (line are translated as entered).	Break the line into two lines. If reading from a file, ensure that the file contains only legal BASIC program lines.

LOG DEVICE ERROR

BATCH	F	A hardware error was reported during an output operation on the log device.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?LP NOT FND?			
DUMP	F	A line printer handler was not available on the system.	Check for a typing error in the command line. Indicate a specific, legal output device and filename.
?LTL			
BASIC	F	The line typed was longer than 132 characters.	The line buffer overflows. Break the line into two or more lines, or if reading from a file, ensure that the file contains only legal BASIC program lines.
LU			
BATCH	F	A Lock Up occurred in the BATCH handler because it could not find a free channel or a channel it could use. (This can happen if all 16 channels are opened for magtape or cassette operations within the BATCH stream.)	Check the program and ensure that one channel is left open or is not used by magtape or cassette.
******* M*			
FORTRAN	IN	Multiple defined label.	The label is ignored. Correct the program and recompile.
MACRO	IN	Multiple definition of a symbol. A symbol or label (or local label) was encountered which was equivalent (in the first six characters) to a previously encountered symbol or label.	Correct the program so that all symbols are unique.
M			
MACRO	IN	Multiple definition of a symbol. A symbol or label (or local label) was encountered which was equivalent (in the first six characters) to a previously encountered symbol or label.	Correct the program so that all symbols are unique.

?/M ODD VAL?			
LINKER	F	An odd value was specified for the stack address.	Check for a typing error in the command line. Reenter the command specifying an even value to the /M switch.
?M-BAD FETCH xx			
MONITOR	F	Either an error occurred while reading a device handler from SY, or the address at which the handler was to be loaded was illegal. xx represents the address plus 2 of the location where the error occurred.	Ensure that a .FETCH is not executed from the foreground, and that all handlers used by the foreground are first loaded. Check that the address at which the handler is to be loaded is not out of the bounds of the program, and that the handler is not so large that it would overflow the program bounds; in this case try to allow more space for the handler.
?M-DIR IO ERR xx			
MONITOR	F	An error occurred doing I/O in the directory of a device. xx represents the address plus 2 of the location where the error occurred.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
?M-DIR OVFLO xx			
MONITOR	F	No more directory segments were available for expansion (occurred during file creation via .ENTER). xx represents the address plus 2 of the location where the error occurred.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.
?M-DIR UNSAFE xx			
MONITOR	F	In F/B only, this message appears in addition to any of the other M- diagnostics listed. It indicates that the error occurred while the	The monitor attempts to complete the directory operation before aborting the job. Examine the device directory. Check the procedures for hard error

			USR was updating a device directory and warns that the last directory may have failed. One or more files on that device may be lost. xx represents the address plus 2 of the location where the error occurred.	conditions listed in Section 1.0 at the beginning of this manual.
?M-FP TRAP xx	MONITOR	F	A floating-point exception trap occurred, and the user program had no .SFPA execption routine active. xx represents the address plus 2 of location where the error occurred.	The job is aborted. Examine the data for floating-point overflow or underflow and adjust it accordingly.
?M-ILL ADDR xx	MONITOR	F	Under the F/B Monitor, an address specified in a monitor call was odd or was not within the job's limits. xx represents the address plus 2 of the location where the error occurred.	Correct the address in error in the source program.
?M-ILL CHAN xx	MONITOR	F	A channel number that was too large was specified. xx represents the address plus 2 of the location where the error occurred.	The default is 16 channels. Define a larger number (255 maximum) using the .CDFN request.
?M-ILL EMT xx	MONITOR	F	An EMT that did not exist was executed (i.e., the function code was out of bounds). xx represents the address plus 2 of the location where the error occurred.	Check the EMT instruction to determine the correct code.

?M-ILL USR xx

MONITOR	F	The USR was called from a completion routine. (This error does not have a soft return, i.e., .SERR will not inhibit this message; see Chapter 9 of the <u>RT-11 System Reference Manual</u>, DEC-11-ORUGA-C-D). xx represents the address plus 2 of the location where the error occurred.	Correct the program; the USR cannot be called from within a completion routine.

?M-NO DEV xx

MONITOR	F	A READ/WRITE operation was attempted but no device handler was in memory for it. xx represents the address plus 2 of the location where the error occurred.	Verify that no .RELEASE was done before the READ/WRITE operation. LOAD the appropriate handler before running the program.

?M-OVLY ERR xx

MONITOR	F	A user program with overlays failed to read an overlay. xx represents the address plus 2 of the location where the error occurred.	Verify that the device is not off-line and that the proper handler is loaded if the overlay program is running from a device type other than that of the system device.

?M-SWAP ERR

MONITOR	F	A hard error was reported while the system was attempting to write a user program to the system swap blocks. This may indicate that the system device is write-locked.	Verify that the system device is write-enabled. Check the procedures for hard error conditions listed in Section 1.0 the beginning of this manual. The error condition will cause the system to halt under the Single-Job Monitor.

?M-SYS ERR

MONITOR	F	An I/O error occurred while trying to read KMON/USR into memory, indicating that the monitor file is situated on the system device in an area that has developed one or more bad blocks. The monitor prints the message and loops trying to read KMON. The message is a warning that the system device is bad.	If, after several seconds, it is apparent that attempts to read KMON are failing, halt the processor. It may be impossible to boot the volume because of the bad area in the monitor file. Use another system device to verify the bad blocks and follow the recovery procedures described for the PIP /K switch in the RT-11 System Reference Manual, DEC-11-ORUGA-C-D. This error condition will cause the system to halt under the Single-Job Monitor.

?M-TRAP TO 4 xx
?M-TRAP TO 1Ø xx

MONITOR	F	The job referenced illegal memory or device registers, or an illegal instruction was used. Stack overflow occurred, a word instruction was executed with an odd address, or a hardware problem caused bus time-out traps through location 4. xx indicates the location where the failure occurred.	Determine the bounds of the user program from the link map or absolute locations 40, 46, 50 and 54. If the error occurred within the bounds of the user program, correct the programming logic. Verify that the program has not corrupted vital monitor data, such as the stack, the queue elements, or the monitor itself. Check USR swapping and program overlaying for possible errors. Refer to Section 2.0, System Failures, at the beginning of this manual for more information. If none of these errors can be identified, report the problem to Software Communications using an SPR (Software Performance Report); include a program listing and a machine readable source program, if possible.

MACRO ALREADY DEFINED

EXPAND	IN	A macro was defined more than once.	Correct the program logic.

MACRO(S) NOT FOUND

EXPAND	IN	Macros listed in an .MCALL statement were not found in SYSMAC.8K.	Ensure that SYSMAC.8K is present on the system.

?MAP FILE ERROR?

LINKER	F	The device was full or a hardware error was reported for the device specified for map output.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Check the procedures for hard error conditions listed in Section 1.0.

?MBOOT.BOT NOT FOUND ON SY:?

MTINIT	F	The file MBOOT.BCT was not found on the system device.	Verify that the file MBOOT.BCT is on the system device (SY:) before running MTINIT. Transfer it (using PIP) to SY: if it is not.

'$' MISSING

BATCH	F	A $ was not present in the first position of the command line (or card column 1).	Ensure that $ occurs in the command line where expected.

MISSING ASSIGNMENT OPERATOR

FORTRAN	IN	The first operator seen in an arithmetic assignment statement was not an equal sign (=). (Example: I+J=K.)	Correct the arithmetic assignment statement in error.

MISSING COMMA

FORTRAN	IN	The comma delimiter was expected but was not found.	Check the format of the statement in error.

MISSING COMMA IN MACRO ARG			
EXPAND	IN	Spaces or tabs were found within a macro argument when a comma was expected.	The macro expansion is aborted. Use brackets around any arguments containing spaces or tabs.
MISSING DELIMITER IN EXPRESSION			
FORTRAN	IN	Two operands were placed next to each other in an expression, with no operator between them.	Correct the statement in error.
MISSING DOT			
EXPAND	IN	A macro name or argument name did not begin with a dot.	Correct the line in error.
?MISSING END IN MACRO?			
EXPAND	F	The end of input was encountered while storing a macro definition.	Verify that the macro has a corresponding .ENDM statement, and that the macro is not split over two files.
MISSING LABEL			
FORTRAN	IN	A statement label was expected but not found (for example, ASSIGN J TO J). A valid statement label reference should precede 'TO' but does not.	Verify that the reference preceding 'TO' is a valid statement label of an executable statement in the same program unit as the ASSIGN statement.
MISSING QUOTATION MARK			
FORTRAN	IN	In a FIND statement, the logical unit number and record number were not separated by a single quotation mark.	Correct the format of the statement in error.

MISSING RIGHT PARENTHESIS

FORTRAN	IN	A right parenthesis was expected but not found (for example, READ (5,100,); in this case, the right parenthesis was expected where the second comma appears).	Correct the format of the statement in error.

MISSING VARIABLE

FORTRAN	IN	A variable was expected but not found (for example, ASSIGN 100 TO 1).	Verfiy that the reference following 'TO' is a vadid integer variable name.

MISSING VARIABLE OR CONSTANT

FORTRAN	IN	An operand (variable or constant) was expected but a delimiter (comma, parenthesis, etc.) was found instead (for example, WRITE()).	Correct the format of the statement in error.

filnam MNT CAS

CBUILD	A	End-of-tape was encountered during a multi-volume cassette transfer. (filnam represents the name of the continued file.)	To continue the transfer, mount a new cassette and type Y followed by a carriage return. To abort the operation, type N and a carriage return.

MNT CAS

CBUILD	A	An output transfer was attempted to a full cassette.	To continue the transfer, mount a new cassette and type a carriage return. To halt the transfer, type N and a carriage return; that portion of the file already on the full cassette will be deleted.

MOD NAME:
mod1 <CR>
mod2 <CR>
<CR>

LIBR A This message is printed following use of the Librarian /D switch. (mod1, mod2, etc. represent module names to be entered by the user).

Enter the names of the modules to be deleted from the Library, following each with a carriage return. A single carriage return terminates the entire list.

MODES OF EXPRESSION MUST BE INTEGER

FORTRAN IN The initial, terminal, or incremental parameters of a DO statement were other than integer expressions. (Only the first term in the expression is checked.)

Correct the DO statement in error.

*MODES OF VARIABLE **** AND DATA ITEM DIFFER*

FORTRAN IN The data type of each variable and its associated data list item must agree in a DATA statement. (**** represents the variable.)

Correct the format of the items in the DATA statement.

?MODULE NOT FOUND?

PATCHO IN The module requested in a POINT command was not found in the input file between the position of the file at the time of the POINT and the file's end.

Check for a typing error in the command line. Verify that the correct module name was entered. Check the order in which the modules were combined; try searching for the object module from the beginning of the file.

?MORE THAN 5 CSECTS REQUIRE CHANGE?

PATCHO F An attempt was made to patch locations in too many different CSECTs.

Make the patch in several steps.

?MORE THAN 15 CHANGES?

PATCHO	F	Too many changes were specified for a particular module.	Make the patch in several steps.

MTn:/ZERO/BOOT-ARE YOU SURE?

MTINIT	A	Confirmation must be given by the user before the magtape can be zeroed.	Verify that the device indicated in the command line is the correct one and confirm the operation (Y zeroes the magtape; any other character aborts the command).

MULT DEF OF xxxxxx

LINKER	W	The symbol (xxxxxx) was defined more than once.	Extra definitions are ignored.

*MULTIPLE DECLARATION FOR VARIABLE *****

FORTRAN	IN	A variable appeared in more than one data type declaration statement or dimensioning statement. (**** represents the variable name.)	Correct the program logic.

MULTIPLE SWITCH

BATCH	F	The same switch was specified more than once in a single command line.	Correct the command line.

?MUST OPEN WORD?

PATCH	F	The @ command was typed when a byte location was open.	Check for a typing error in the command line. Use the @ command only when a word is open.

?MUST SPECIFY SEG?

PATCH	F	The address referenced was not in the root section.	Check for a typing error in the command line. Check the Linker load map and specify a segment number using S:.

******* N*			
MACRO	IN	A number containing an 8 or 9 was missing a decimal point.	Ensure that numbers that are intended to be decimal values are followed by a decimal point.
N			
MACRO	IN	A number containing an 8 or 9 was missing a decimal point.	Ensure that numbers that are intended to be decimal values are followed by a decimal point.
NAME DOESN'T MATCH			
EXPAND	IN	The optional name specified in an .ENDM directive did not match the name given in the corresponding .MACRO directive.	Correct the .ENDM statement.
?NBF AT LINE xxxxx			
BASIC	F	The NEXT statement corresponding to a FOR statement preceded the FOR statement.	Ensure that a FOR statement occurs before the NEXT statement.
?NER AT LINE xxxxx			
BASIC	F	There was not enough room on the selected device for the specified number of output blocks.	Specify fewer blocks in the OPEN statements, or select another device.
NESTED MACROS			
EXPAND	IN	A macro was defined or invoked within another macro.	Correct the program so that it contains no nested macros.
NEXT BEFORE FOR AT LINE xxxxx			
BASIC	F	The NEXT statement corresponding to a FOR statement preceded the FOR statement.	Ensure that a FOR statement occurs before the NEXT statement.

?NIF			
ASEMBL	F	No input file was specified.	Reenter the command, specifying at least one input file.
?NO ADDR OPEN?			
PATCH	F	The <line feed>, ^ or @ command was typed when no location was open.	Check for a typing error in the command line. Open a location before using these commands.
?NO CLOCK?			
MONITOR	F	No KW11L clock was available for the TIME command.	Enter another command. The TIME command cannot be used.
NO CONTROL FILE			
BATCH	F	An attempt was made to send the .CTL file to a nonfile-structured device (e.g., LP).	Check for a typing error in the command line. Use a file-structured device for the CTL file, or use the /N switch to inhibit execution.
?NO CORE?			
LIBR	F	Available free memory has been used up.	The current command is aborted. Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
NO $EOJ			
BATCH	F	A $JOB or $SEQUENCE command appeared without a preceding $EOJ to end the previous job.	Correct the BATCH stream by inserting an $EOJ command.
?NO FG?			
MONITOR	F	A SUSPEND, RSUME, or UNLOAD FG command was given, but no foreground job was in memory.	Check for a typing error in the command line. Enter a command that does not require a foreground job.

*?*NO FILE*?*			
EDIT	F	An attempt was made to read or write when no file was open.	Check for a typing error in the command line. Open a file first before attempting an I/O operation.
NO FILE			
BATCH	F	No file descriptor was found where expected in the BATCH stream, or no file name was entered in the $CREATE command.	Enter a filename where expected.
NO FILE NAME BEFORE "."			
BATCH	F	An extension was specified but no file name preceded it.	Correct the format of the file descriptor.
?NO FILE OPEN?			
PATCHO	F	An attempt was made to use a command other than DEC or HELP before an OPEN command was issued.	Check for a typing error in the command line. A file must be open before any commands other than DEC or HELP can be issued.
?NO INPUT?			
LINKER	F	No input file was specified; there must be at least one object module specified as input.	Check for a typing error in the command line; ensure that an object module is specified as input; correct the command line.
?NO INPUT FILE?			
EXPAND MACRO	F F	No input file was specified; there must be at least one input file.	Check for a typing error in the command line. Correct the command line.
NO LOGICAL DEVICE			
BATCH	F	No logical device was specifi in a $MOUNT command.	Correct the command format.

NO NAME

EXPAND	IN	A macro definition had no name.	Correct the line in error.

NO PHYSICAL DEVICE

BATCH	F	No physical device was specified in a $MOUNT command.	Correct the command format.

NO PROGRAM AT LINE xxxxx

BASIC	F	The RUN command was specified, but no program was in memory.	Type in a program or restore a previously saved program using the OLD command.

*?*NO ROOM*?*

EDIT	F	An attempt was made to Insert, Save, Unsave, Read, Next, Change or Exchange when there was not enough room in the appropriate buffer.	Delete unwanted buffers to create more room (using the 0U or 0M commands; see Chapter 3 of the <u>RT-11 System Reference Manual</u>, DEC-11-ORUGA-C-D), or write text to the output file.

?NO SYS ACTION?

CBUILD	IN	The /Y switch was not included with a command specified on a .SYS file.	The .SYS file operations are not performed, but the remainder of the command is executed.

?NO .SYS/.BAD ACTION?

MBUILD PIP	IN IN	The /Y switch was not included with a command specified on a .SYS or .BAD file. A "wild-card" (*.*) transfer is most likely to cause this message.	Check for a typing error in the command line. The command is executed for all but .SYS and .BAD fies. Use /Y if .SYS and .BAD files should also be included in the operation.

*?*NO TEXT*?*

EDIT	F	An attempt was made to call in text from the Save Buffer when there was no text available.	Check for a typing error in the command line. Use the Save command to store text in the Save Buffer, before calling it using the Unsave command.

?NO UFD?

FILEX	F	The specified UFD was not found on the DOS input disk.	Verify that no typing error has been made, and that the input disk is the correct one.

NON-STANDARD STATEMENT ORDERING

FORTRAN	W	The message warns that non-adherence to statement ordering requirements may cause error conditions under other FORTRAN compilers.	Check the PDP-11 FORTRAN Language Reference Manual (DEC-11-LFLRA-B-D) for statement ordering requirements and to determine if any changes to the program should be made.

NOT ENOUGH ROOM AT LINE xxxxx

BASIC	F	There was not enough room on the selected device for the specified number of output blocks.	Specify fewer blocks in the OPEN statements, or select another device.

?NOT IN PROGR BOUNDS?

PATCH	F	An attempt was made to open a location beyond the end of the file. The value of the initial stack pointer for the program may also be beyond the last location of the program.	Check for a typing error in the command line. Check the Linker load map to determine where the program was loaded. Check the initial value of the stack pointer.

?NPR AT LINE xxxxx

BASIC	F	The RUN command was specified, but no program was in memory.	Type in a program or restore a previously saved program using the OLD command.

?NSM AT LINE xxxxx

BASIC	F	String and numeric variables were found in the same expression or were set equal to each other (as in A$=2).	Ensure that string expressions are assigned to string variables.

NUMBERS IN FORMAT STATEMENT NOT IN RANGE

FORTRAN	IN	An integer constant in a FORMAT statement was not in the proper range.	Correct the statement so that the integer constant falls within the legal range (1 to 255).

NUMBER AND STRINGS MIXED AT LINE xxxxx

BASIC	F	String and numeric variables were found in the same expression or were set equal to each other (as in A$=2).	Ensure that string expressions are assigned to string variables.

******* O*

MACRC	IN	Opcode error. The directive was out of context.	Correct the program.

O

FORTRAN	F	An unrecoverable error occurred while the Compiler was writing the object file (.OBJ), or there was possibly insufficient output file space.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Check the procedures for hard error conditions listed in Section 1.0.
MACRO	IN	Opcode error. The directive was out of context.	Correct the program.

/O IGNORED

LINKER	W	Overlays have been specified in the wrong order.	Check for a typing error in the command line. The overlay switch is ignored. Consult the overlay restrictions in the Linker chapter of the RT-11 System Reference Manual (DEC-11-ORUGA-C-D).

MBUILD PIP	F F	An illegal output file specification was given, or a required output file was missing from the command line.	Check for a typing error in the command line. Ensure that /X is included in any "wild-card" (*.*) construction used in a copy operation, and that the dev:filnam.ext specification is in the correct format.

OUT OF DATA AT LINE xxxxx

BASIC	F	The data list was exhausted and a READ requested additional data.	Add additional data items or reuse data by executing the RESTORE statement.

?OUTPUT DEVICE FULL?

EXPAND MACRO	F F	There was no room to continue writing output.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

OUTPUT DEVICE FULL

BATCH	F	The temporary file (.CTL) created by BATCH was too large for the specified device.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

?OUTPUT ERROR?

PATCHO	F	A hard error was reported while writing the output file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

OUTPUT ERROR

BATCH	F	Magtape or cassette was specified as the .CTL output device. A hard error was reported while BATCH was attempting to write the compiler output file (.CTL).	Check for a typing error in the command line. Verify that the output device specified is neither magtape nor cassette. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

OUTPUT FILE NOT OPEN

BATCH	F	The .CTL file output channel was not opened. This is most likely due to an error in the BATCH compiler.	Report the problem to Software Communications using an SPR (Software Performance Report); include a program listing and a machine readable source program, if possible.

?OUTPUT FILE TOO SMALL?

PATCHO	F	The space allocated to the output file was too small.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

?OUTPUT FULL?

LINKER	F	The output device was full.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

OVERFLOW AT LINE xxxxx

BASIC	F	The result of a computation was outside the legal range of floating point numbers.	Check the program logic.

?OVF AT LINE xxxxx

BASIC	F	The result of a computation was outside the legal range of floating point numbers.	Check the program logic.

?OVR COR?

MONITOR	F	An attempt was made to GET or RUN a file that was too big.	Check for a typing error in the command line. Refer to Section 3.0, at the beginning of this manual, for information on how to increase storage space.

filnam ?OVRFL?

CBUILD	F	During a nonmulti-volume transfer, the file being transferred was too large for a single cassette.	The partial file is deleted from the cassette; the transfer continues with the next file, if any.

?OVRFL?

CBUILD	F	The file did not fit on the cassette.	The file is not transferred. Use another cassette with more room for files.

******* p*

FORTRAN	IN	A statement contained unbalanced parentheses.	The statement is aborted. Correct the statement in error and recompile.
MACRO	IN	Phase error. A label's definition or value varied from one pass to another, or a local symbol occurred twice within a local symbol block.	Correct the program logic.

P

FORTRAN	F	Optimizer push down overflow. The statement was too complex, or too many common subexpressions occurred in one basic block of a program.	Simplify complex statements.
MACRO	IN	Phase error. A label's definition or value varied from one pass to another, or a local symbol occurred twice within a local symbol block.	Correct the program logic.

P-SCALE FACTOR NOT IN RANGE -127 to +127

FORTRAN	IN	P-scale factors were not in the correct range.	Correct the statement in error. The P-scale factor must be a signed or unsigned integer constant in the range -127 to +127.

?PARAMS?

MONITOR	F	Bad parameters were typed to the SAVE command.	Check for a typing error in the command line. Check the format of the SAVE command and reenter the command.

PARENTHESES NESTED TOO DEEPLY

FORTRAN	IN	Group repeats in a FORMAT statement were nested too deeply.	Limit group repeats to eight levels of nesting.

PAUSE -- ?BAD PATCH?

PATCHO	A	The checksum entered on exit did not agree with that calculated by PATCHO.	Ensure that the correct module is being patched and that no typing error occurred while entering either the patch or the checksum. PATCHO will pause, waiting for a user-response. Type a CTRL C to abort PATCHO, without making the patch permanent. Type a carriage return to allow the patch to be made as specified.

PLEASE ASSIGN LOG,LST

BATCH	F	The log device (LOG:) and/or the listing device (LST:) were not assigned.	Use the ASSIGN command to assign LOG: and LST: devices. LOG: must be assigned, and LST: is recommended unless it is certain the BATCH stream will not reference LST:.

PLEASE LOAD LOG HANDLER

BATCH	F	The log device handler was not resident.	Check for a typing error in the LOAD command. Load the appropriate device handler.

?PROG HAS NO SEGS?

PATCH	F	The file specified as an overlay file was not.	Check the Linker load map. Verify that the correct file was specified in the command line.

PROGRAM TOO BIG AT LINE xxxxx

BASIC	F	The line just entered caused the program to exceed the user code area.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines; Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Reduce the size of arrays in memory to the size required (DIMension statement); Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

?PTB AT LINE xxxxx

BASIC	F	The line just entered caused the program to exceed the user code area.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines; Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Reduce the size of arrays in memory to the size required (DIMension statement); Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

******* Q*

MACRO	IN	Questionable syntax. There were missing arguments, the instruction scan was not completed, or a carriage return was not immediately followed by a line feed or form feed.	Correct the program.

Q

MACRO	IN	Questionable syntax. There were missing arguments, the instruction scan was not completed, or a carriage return was not immediately followed by a line feed or form feed.	Correct the program.

******* R*

MACRO	IN	Register-type error. An invalid use of or reference to a register was made.	Correct the statement in error.

R

FORTRAN	F	A hard error was reported while the Compiler was reading the source file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
MACRO	IN	Register-type error. An invalid use of or reference to a register was made.	Correct the statement in error.

?RBG AT LINE nnnnn

BASIC	F	A RETURN was encountered before execution of a GOSUB statement.	Ensure that subroutines are entered only through a GOSUB statement.

?READ ERROR?

PATCH	F	A hard error was reported during a read operation.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?REBOOT?

CBUILD MBUILD PIP	IN IN IN	.SYS files have been transferred, renamed, compressed or deleted from the system device.	If any of the .SYS files in use by the current system (MONITR.SYS and handler files) have been physically moved on the system device, it is necessary to reboot the system immediately. (The actual reboot operation must not be performed until PIP returns with the prompting asterisk for the next command.) Otherwise, the message can be ignored. If /O is used to reboot, ensure that the original system device is mounted. If the cause of the message was a /S operation, a manual reboot must be done.

*REFERENCE TO INCORRECT TYPE OF LABEL *****

FORTRAN	IN	A statement label reference that should be a label on a FORMAT statement was not such a label, or a statement label reference that should be a label on an executable statement was not such a label. (**** represents the label name.)	Correct the program logic.

REFERENCE TO UNDEFINED STATEMENT LABEL

FORTRAN	IN	Reference was made to a statement number that was not defined anywhere in the program unit.	Correct the program logic.

?REL FIL I/O ER?

MONITOR	F	Either the file requested was not a REL file, or a hardware error was encountered trying to read or write the file.	Relink the program using the /R switch to produce a REL file before trying to FRUN it. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?REL FILE ERR?

LINKER	F	A hard error was reported while writing a REL file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

RETURN BEFORE GOSUB AT LINE xxxxx

BASIC	F	A RETURN was encountered before execution of a GOSUB statement.	Ensure that subroutines are entered only through a GOSUB statement.

RETURN FROM CALL ERROR

BATCH	F	BATCH could not read the control file that called a subprogram. Execution could not be resumed on return from the call.	Verify that the .CTL file has not been destroyed; recompile if necessary. Retry the operation. Use another drive or unit if possible.

?ROOM?

PIP	F	There was insufficient space following the file specified with a /T switch.	Check for a typing error in the command line. Use the /E switch to obtain a directory listing and verify that sufficient space follows the file.

?RPL AT LINE xxxxx

BASIC	F	The file specified in a SAVE command already existed.	Use a different name in the SAVE command or use the REPLACE command.

******* S*

FORTRAN	IN	Syntax error. The statement was not of the general FORTRAN statement form.	The statement is aborted. Correct the error and recompile.

S

FORTRAN	F	Subexpression stack overflow; the statement was too complex.	Simplify complex statements.

?SAV FILE ERR?			
LINKER	F	A hard error was reported while writing a save image file.	Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.
SEPARATOR MISSING			
BATCH	F	A file descriptor was not terminated by a space, a +, a comma, or a carriage return.	Correct the format of the file descriptor.
?SUB AT LINE xxxxx			
BASIC	F	The subscript computed was greater than 32,767 or was outside the bounds defined in the DIM statement.	Ensure that array subscripts fall within the legal range.
*?*SRCH FAIL*?*			
EDIT	F	The text string specified in a Get, Find or Position command was not found in the available data.	Check for a typing error in the command line. Ensure that the text string exists in the file as specified in the command line. If this error occurs following a Get command, the pointer is positioned at the end of the current Text Buffer and the command may be reentered. After a Find or Position command, End of File is detected and the pointer is positioned at the beginning of an empty Text Buffer; close the file, reopen it, and reenter the command.

?SSO AT LINE xxxxx

BASIC	F	There was not enough memory available to store all the strings used in the program.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines; Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Reduce the size of arrays in memory to the size required (DIMension statement); Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

STACK ADDRESS = sym

LINKER	A	This message is printed following use of the Linker /M switch in which an octal value was not specified. (sym represents a global symbol to be entered at the keyboard.)	Specify the global symbol whose value is to be the initial stack address.

?STACK ADDRESS UNDEFINED OR IN OVERLAY?

LINKER	W	The stack address specified by the /M switch was either undefined or in an overlay.	Check for a typing error in the command line. The stack address is set to the system default (1000 for .SAV files, 0 for .REL files).

STATEMENT MUST BE UNLABELED

FORTRAN	IN	A DATA, SUBROUTINE, FUNCTION, BLOCK DATA, arithmetic statement function definition, or declarative statement was labeled.	Correct the format of the statement in error.

STATEMENT TOO COMPLEX

FORTRAN	IN	An arithmetic statement function had too many dummy arguments, or the statement was too long to compile.	Verify that the number of dummy arguments in an arithmetic statement does not exceed ten; break long statements into two or more smaller statements.

?STL AT LINE xxxxx

BASIC	F	The length of a string in a BASIC statement exceeded 255 characters.	Reduce the size of the string. The maximum length allowed is 255 characters.

STRING STORAGE OVERFLOW AT LINE xxxxx

BASIC	F	There was not enough memory available to store all the strings used in the program.	Reduce the program size. This can be done by one of several procedures: Eliminate or reduce unnecessary items such as REMark statements, long printed messages, and optional keywords such as LET; Make maximum use of multiple statement lines; Make efficient use of program loops, subroutines, and user-defined functions; Split up large programs into several smaller programs by use of CHAIN or OVERLAY statements; Use virtual arrays in memory to the size required (DIMension statement); Use virtual array files for arrays that are too large to fit into memory; Reduce the number of variables and arrays in a program by reusing them when their contents are no longer needed, instead of creating new variables or arrays; Reduce the number of simultaneously open files by opening a file just before it is needed and closing it immediately after the last use. After program lines are deleted the program can be stored by the SAVE command and restored by the OLD command to further optimize program memory requirements.

STRING TOO LONG AT LINE xxxxx

BASIC	F	The length of a string in a BASIC statement exceeded 255 characters.	Reduce the size of the string. The maximum length allowed is 255 characters.

SUBROUTINE OR FUNCTION STATEMENT MUST BE FIRST

FORTRAN	IN	A SUBROUTINE, FUNCTION or BLOCK DATA statement was not the first statement in a program unit.	Ensure that, if present, these statements appear first in a program unit.

*SUBSCRIPT OF ARRAY **** NOT IN RANGE*

FORTRAN	IN	Array subscripts that are constants or constant expressions were found to be outside the bounds of the array's dimensions. (**** represents the array name.)	The operation in question is aborted. Correct the program.

SUBSCRIPT OUT OF BOUNDS AT LINE xxxxx

BASIC	F	The subscript computed was greater than 32,767 or was outside the bounds defined in the DIM statement.	Ensure that array subscripts fall within the legal range.

?SV FIL I/O ER?

MONITOR	F	An error was reported for a .SAV file during a SAVE (output) or R, RUN, or GET (input) command. Possible errors include end-of-file, hard error, and channel not open.	Verify that the file specified is a legal .SAV file or that enough room existed on the device during the SAVE. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?SWITCH ERROR?

SRCCOM	F	An invalid switch was found or a switch other than /L was given a value.	Check for a typing error in the command line. Use only those switches listed in Table K-1 of the RT-11 System Reference Manual (DEC-11-ORUGA-C-D) for SRCCOM.

SWITCH NOT UNIQUE

BATCH	F	The spelling of a switch was not unique (appears only when $JOB UNIQUE is specified).	Correct the spelling so that the switch contains enough characters to be unique.

?SWT ERR?

FILEX	F	An attempt was made to use more than one /S or /T switch in a command line (only one is allowed); an attempt was made to use more than one transfer mode switch (/I, /P, /A) or more than one operation switch (/D, /L, /F, /Z) in a command line (only one of each is allowed).	Check for a typing error in the command line. Reenter the command line specifying the correct number of switches.

?SY I/O ER?

MONITOR	F	A hard error condition was reported on the system device while reading or writing the swap area.	Verify that the system device is write-enabled. Check the procedures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?SYMBOL TABLE OVERFLOW?

LINKER	F	There were too many global symbols used in the program.	Retry the link using the /S switch. If the error still occurs, the link cannot take place in the available memory. Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.

?SYN AT LINE xxxxxx

BASIC	F	The program encountered an unrecognizable statement. Common examples of syntax errors are misspelled commands, unmatched parentheses, and other typographical errors. The wrong number of arguments or an illegal delimiter in a function can also cause this error.	Check for a typing error in immediate mode commands. Correct the program statement.

SYNTAX

EXPAND	IN	A macro directive was not constructed correctly.	Correct the format of the macro directive.

SYNTAX ERROR

FORTRAN	IN	The statement was not of the general FORTRAN statment form.	Check the general format of the statement in error and correct the program.

SYNTAX ERROR AT LINE xxxxxx

BASIC	F	The program encountered an unrecognizable statement. Common examples of syntax errors are misspelled commands, unmatched parentheses, and other typographical errors. The wrong number of arguments or an illegal delimiter in a function can also cause this error.	Check for a typing error in immediate mode commands. Correct the program statement.

SYNTAX ERROR IN INTEGER OR FLOATING CONSTANT

FORTRAN	IN	An integer or floating constant was incorrectly formed (for example, 1.23.4).	Correct the format of the integer or floating constant in question.

SYSLIB-FATAL INTERRUPT OVERRUN

SYSLIB	F	Interrupt overrun occurred (i.e., the interrupt routine had control for a period of time longer than the time in which two more interrupts using the same vector occurred).	The job is aborted. Tasks requiring very fast interrupt response may not be able to run under FORTRAN.

***** *T*

MACRO	IN	Truncation error. A number generated more than 16 bits of significance or an expression generated more than 8 bits of significance during the use of the .BYTE directive.	A number is truncated to 16 bits and an expression to 8 bits of significance. Correct the source program.

T

FORTRAN	F	Memory overflow.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space.
MACRO	IN	Truncation error. A number generated more than 16 bits of significance or an expression generated more than 8 bits of significance during the use of the .BYTE directive.	A number is truncated to 16 bits and an expression to 8 bits of significance. Correct the source program.

?TTM?

MONITOR	F	The TIME command argument was illegal.	Check for a typing error in the command line. Reenter the TIME command using the correct format.

?TIM ERROR?

CBUILD	F	A timing error occurred, probably due to a cassette hardware error.	Check the procudures for hard error conditions listed in Section 1.0 at the beginning of this manual.

?TLT AT LINE xxxxxx

BASIC	F	The line just entered exceeded the area available for translation (lines are translated as entered).	Break the line into two lines. If reading from a file, ensure that the file contains only legal BASIC program lines.

?TMO?

ASEMBL	F	Too many output files were specified.	Check for a typing error in the command line. Enter only an object and/or listing file as output.

TOO MANY ARGS

EXPAND	IN	A macro directive had too many arguments.	Ensure that a macro directive has no more than 30 arguments.

TOO MANY FILE DESCRIPTORS

BATCH	F	More than six file descriptors were specified in a $command line.	Check for a typing error in the command line. Limit the number of file descriptors to six. Check the format of the command in question.

?TOO MANY OUTPUT FILES?

LINKER	F	Too many output files were specified.	Check for typing error in the command line. Limit the number of output file specifications to two (.SAV or .REL output file, and the link map).

TOO MANY OUTPUT FILES

BATCH MACRO	F F	Too many output files were specified.	Check for a typing error in the command line. Limit the number of output file specifications to two. (For BATCH, output files represent the compiler output device and file and the log file; for MACRO, the output files represent the object output and listing.)

?TOO MUCH DIFFERENCE?

SRCCOM	F	More than 310 (octal) lines of difference between two files were found.	Check for a typing error in the command line. A limit of 310 octal lines of difference is set for SRCCOM (note that the value specified to the /L switch must be <=310).

TRANSFER ADDRESS =
sym <CR>

LINKER	A	This message is printed whenever the Linker /T switch is followed by a carriage return rather than an octal value. (sym represents a global symbol.)	Specify the global symbol whose value is the transfer address of the load module.

TRANSFER ADDRESS UNDEFINED OR IN OVERLAY

LINKER	W	The transfer address was not defined or was in an overlay.	Check for a typing error in the command line. The response to the /T switch must be either a colon followed by an unsigned 6-digit octal number, or a carriage return followed by the global symbol whose value is the transfer address of the load module.

******* U*

FORTRAN	IN	The statement could not be identified as any legal FORTRAN statement.	The statement is aborted. Check for incorrect spelling.
MACRO	IN	Undefined symbol. An undefined symbol was encountered during the evaluation of an expression.	Relative to the expression, the undefined symbol is assigned a value of zero. Correct the source program.

U

MACRO	IN	Undefined symbol. An undefined symbol was encountered during the evaluation of an expression.	Relative to the expression, the undefined symbol is assigned a value of zero. Correct the source program.

?UFN AT LINE xxxxxx

BASIC	F	The function called was not defined by the program or was not loaded with BASIC.	A user-defined function cannot be defined by an immediate mode statement. Check that the name of the routine in the CALL statement is correctly spelled.

?ULN AT LINE xxxxxx

BASIC	F	The line number specified in an IF, GO TO or GOSUB statement did not exist anywhere in the program.	Check the program logic.

UNDEF GLBLS

LINKER	W	A load map was requested and undefined globals existed.	Check for a typing error in the command line. The undefined globals are listed in the link map. Correct the source program. Verify that all necessary object modules are indicated in the command line or present in the libraries specified.

UNDEFINED FUNCTION AT LINE xxxxxx

BASIC	F	The function called was not defined by the program or was not loaded with BASIC.	A user-defined function cannot be defined by an immediate mode statement. Check that the name of the routine in the CALL statement is correctly spelled.

UNDEFINED GLOBALS
xxxxxxx
xxxxxxx

LINKER	W	The globals listed (xxxxxx) were undefined.	Check for a typing error in the command line. Correct the source program by defining all globals. Verify that all necessary object modules are indicated in the command line or present in the libraries specified.

UNDEFINED LINE NUMBER AT LINE xxxxx

BASIC	F	The line number specified in an IF, GO TO or GOSUB statement did not exist anywhere in the program.	Check the program logic.

UNKNOWN COMMAND

BATCH	F	The command specified with a $ in character position 1 was not a legal BATCH command.	Verify that the spelling of the command is correct.

UNLABELED FORMAT STATEMENT

FORTRAN	IN	A FORMAT statement was not labeled.	Correct the FORMAT statement in error by assigning it the proper label.

!UNLOAD BA!

EATCH	F	This prompting message is printed when the /U switch is given to the compiler.	Type UNL BA <CR> in response to the dot printed by the Keyboard Monitor.

*USAGE OF VARIABLE **** INVALID*

FORTRAN	IN	An attempt was made to EXTERNAL a common variable, an array variable, or a dummy argument, or an attempt was made to place in COMMON a dummy argument or external name. (**** represents the variable name.)	Correct the progam logic.

USE REPLACE AT LINE xxxxx

BASIC	F	The file already existed.	Use a different name or use the REPLACE command.

VALUE OF CONSTANT NOT IN RANGE

FORTRAN		An integer constant in the designated source program line exceeds the maximum unsigned value (+65536). This may also be the case if an invalid dimension is specified for an array, or if the exponent of a floating point constant is too large.	Correct the program statement in error.

*VARIABLE **** INVALID IN ADJUSTABLE DIMENSION*

FORTRAN	IN	A variable used as an adjustment dimension was not an integer dummy argument in the subprogram unit. (**** represents the variable name.)	Correct the program.

*VARIABLE **** IS NOT WORD ALIGNED*

FORTRAN	W	A non-LOGICAL*1 variable or array was placed after a LOGICAL*1 variable or array in COMMON, or non-LOGICAL*1 variables or arrays were equivalenced to LOGICAL*1 variables or arrays. Attempting to reference NON-LOGICAL variables that are not word-aligned may cause the run-time error ?ERR 61 ILLEGAL MEMORY REFERENCE to be reported. (**** represents the variable name.)	In the case of the run-time error condition, the variables in question must be aligned by modifying the appropriate COMMON statements.

*VARIABLE **** NAME EXCEEDS SIX CHARACTERS*

FORTRAN	W	A variable name of more than six characters was specified.	The first six characters are used as the variable name. (Note that other FORTRAN compilers may treat this as an error condition).

W

FORTRAN	F	An unrecoverable error occurred while the compiler was writing the listing file. Possibly, the listing file space was not large enough.	Refer to Section 3.0, at the beginning of this manual, for information on how to increase memory space. Check the device for a write-locked or off-line condition.

[WARNING] MSG# m TEXT

FORTRAN		These messages indicate a condition that may be potentially dangerous at execution time or that may present compatibility problems with other FORTRAN compilers. m specifies the error number and text is an explanation of the error.	See the individual listings of "text" for a description of each message.

?WLO AT LINE xxxxx

BASIC	F	An attempt was made to write on a sequential or virtual file opened for input only.	Open a sequential file for output, or open a virtual file without indicating FOR INPUT or FOR OUTPUT.

?WRITE ERROR?

PATCH	F	A file I/O error occurred while writing.	Check the procedures for hard error conditions listed at the beginning of this manual.

WRITE FAILED

PTBUILD	F	The system disk was write-protected.	Verify that the system disk is write-enabled. Retry the operation.

?WRT LOCK?

CBUILD	F	The cassette unit specified was write-locked.	Write-enable the cassette by positioning the tab so that the hole is covered.

WRITE LOCKOUT AT LINE xxxxxx

BASIC	F	An attempt was made to write on a sequential or virtual file opened for input only.	Open a sequential file for output, or open a virtual file without indicating FOR INPUT or FOR OUTPUT.

?WRONG NUMBER OF OUTPUT FILES?

EXPAND	F	Too many output files (or none) were specified.	Check for a typing error in the command line. Only one output file is allowed and must be indicated in the command line.

*WRONG NUMBER OF SUBSCRIPTS FOR ARRAY *****

FORTRAN	IN	An array reference did not have the same number of subscripts as specified when the array was dimensioned. (**** represents the array name.)	Correct the statement in error.

Y

FORTRAN	F	Code generation stack overflow. The statement was too complex.	Simplify complex statements.

******* Z*

MACRO	IN	The message indicates an instruction which is not compatible among all members of the PDP-11 family.	Verify that the instruction is legal for the machine in use.

Z

FORTRAN	F	Compiler error.	Report the error to Software Communications using an SPR (Software Performance Report); include a program listing and a machine-readable source program, if possible.

MACRO	IN	The message indicates an instruction which is not compatible among all members of the PDP-11 family.	Verify that the instruction is legal for the machine in use.

dev:/Z ARE YOU SURE?

FILEX PIP	A A	Confirmation must be given by the user before a device can be zeroed.	Verify that the device indicated in the command line is the correct one and confirm the operation; Y zeroes the device (or directory in the case of FILEX); any other character aborts the command.

INDEX

INDEX (Cont.)

RT-11 System Message Manual
DEC-11-ORMEA-A-D

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Repcrt (SPR) form.

Did you find errors in this manual? If so, specify by page.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________

Organization ______________________________

Street ______________________________

City ______________ State ____________ Zip Code or Country ____________

If you requi:e a written reply, please check here. □

Please cut along this line.

-- **Fold Here** --

-- **Do Not Tear - Fold Here and Staple** --

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Communications
P. O. Box F
Maynard, Massachusetts 01754

digital

digital equipment corporation

Printed in U.S.A.