

SC34-0445-1

Event Driven Executive Messages and Codes

Version 4.1

**Library Guide and
Common Index**

SC34-0441

**Installation and
System Generation
Guide**

SC34-0436

**Operator Commands
and
Utilities Reference**

SC34-0444

**Language
Reference**

SC34-0442

**Communications
Guide**

SC34-0443

**Messages and
Codes**

SC34-0445

Operation Guide

SC34-0437

**Event Driven
Language
Programming Guide**

SC34-0438

**Reference
Cards**

SBOF-1211

**Problem
Determination
Guide**

SC34-0439

**Customization
Guide**

SC34-0440

**Internal
Design**

LY34-0246

Second Edition (December 1983)

This is a major revision of, and obsoletes, SC34-0445-0. Technical changes to the text and illustrations are indicated by a vertical line to the left of the changes.

Use this publication only for the purpose stated in the Preface.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to IBM Corporation, Information Development, Department 28B, P. O. Box 1328, Boca Raton, Florida 33432. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Summary of Changes for Version 4.1

The following changes have been made to this document. (Changes are indicated by a vertical line in the left-hand margin of the affected page.)

- New messages have been added to *Chapter 2* which are issued by the following utilities and software features:
 - \$DISKUT2
 - \$EDXASM
 - \$LOG
 - Overlay Manager
 - Session Manager (\$SMMAIN)
 - Spooling
 - Systems Network Architecture (\$SNAMAIN)
 - \$TAPEUT1
 - X.21 Circuit Switched Network
- A list of return codes for the X.21 Circuit Switched Network has been added to *Chapter 5*.
- New return codes also have been added to the following return code lists in *Chapter 5*:
 - Binary Synchronous Communications (BSC)
 - EXIO
 - NETINIT
 - Session Termination (SNA)
 - Spool
 - Terminal I/O – General

Summary of Changes for Version 4.1

- New stop codes have been added to *Chapter 6* for the following modules:
 - BSCINIT
 - INITADAP
 - INIT4980
 - \$OVLMGR0
 - \$SADUMP
 - XPSINIT
- The stop codes have been grouped in numeric order.

About This Book

This book describes the messages and codes issued by the Event Driven Executive Version 4 operating system, and by related licensed programs that run under the Event Driven Executive .

Audience

The information in this book is useful for anyone who uses the Event Driven Executive system, its utility programs or related licensed programs, or who codes, executes, or debugs application programs.

How This Book Is Organized

This book contains two main parts and one appendix:

- “Part 1. Messages” on page MC-1 explains error and special information messages issued by the Event Driven Executive system, utilities, and related licensed programs.
- “Part 2. Codes” on page MC-287 explains completion, post, and return codes issued by the Event Driven Executive system, utilities, and related licensed programs. It also lists the stop codes.
- Appendix A, “Program Checks and Processor Status Word (PSW)” on page MC-379 describes program check error messages, system program check error messages, and the processor status word.

About This Book

How This Book Is Organized (*continued*)

A Guide to the Library

Refer to the *Library Guide and Common Index* for information on the design and structure of the Event Driven Executive Version 4.0 library and for a bibliography of related publications.

Contacting IBM About Problems with Event Driven Executive Services

You can inform IBM of any problems you find when using this book by completing and mailing in the **Reader's Comment Form** at the back of the book.

If you have a problem with the IBM Series/1 Event Driven Executive services, you should fill out an authorized program analysis report (APAR) as described in the *IBM Series/1 Software Service Guide*, GC34-0099.

Contents

Part 1. Messages MC-1

Chapter 1. Introduction to Messages MC-3

How Messages are Documented MC-4

Identifying Variables in Messages MC-4

Finding Messages in the Book MC-5

How Messages are Alphabetized MC-5

Finding Messages that Begin with Special Characters MC-5

Finding Messages that Begin with Numbers MC-6

Other Hints for Finding Messages MC-6

Chapter 2. Messages MC-7

Tips for Finding Messages in This Chapter MC-7

Part 2. Codes MC-287

Chapter 3. Completion Codes MC-289

\$EDXASM Completion Codes MC-290

\$EDXLINK Completion Codes MC-291

\$JOBUTIL Completion Codes MC-292

\$UPDATE Completion Codes MC-293

\$XPSSLINK, \$XPSPOST, \$XPSPRE Completion Codes MC-294

Chapter 4. Post Codes MC-295

BIND Event Post Codes MC-296

Channel Attach Post Codes MC-297

Tape Post Codes MC-299

WAITM Instruction Post Codes MC-300

Contents

Chapter 5. Return Codes MC-301
Binary Synchronous Communications (BSC) Return Codes MC-303
Channel Attach Return Codes MC-305
Data Formatting Return Codes MC-307
Disk and Diskette Read/Write Return Codes MC-308
\$DISKUT3 Return Codes MC-310
EXIO Return Codes MC-311
Floating-Point Return Codes MC-312
Formatted Screen Image Return Codes MC-313
\$IMDATA — Screen Image Unprotected Fields MC-313
\$IMOPEN — Formatted Screen Image MC-313
\$IMPROT — Screen Image Protected Fields MC-313
General Purpose Interface Bus (GPIB) Return Codes MC-314
Indexed Access Method Return Codes MC-315
Job Queue Processor Return Codes MC-318
LOAD Return Codes MC-319
Message Handler Return Codes MC-320
Multiple Terminal Manager Return Codes MC-321
\$PDS Return Codes MC-323
SBIO (Sensor-Based I/O) Return Codes MC-324
SDLC Communications Return Codes MC-325
Series/1-to-Series/1 Attachment Return Codes MC-326
Sort/Merge Return Codes MC-327
Spool Return Codes MC-328
System Return Codes MC-329
Systems Network Architecture Return Codes MC-330
NETCTL Return Codes MC-330
NETGET Return Codes MC-332
NETINIT Return Codes MC-334
NETPUT Return Codes MC-335
NETTERM Return Codes MC-336
Systems Network Architecture Extended Error Codes MC-337
NETBIND Return Codes MC-337
NETCLOSE Return Codes MC-337
NETOPEN Return Codes MC-338
NETRECV Return Codes MC-338
NETSEND Return Codes MC-339
NETUBND Return Codes MC-341
Session Termination Return Codes MC-342
Tape (READ/WRITE) Return Codes MC-343
Terminal I/O Return Codes MC-344
Terminal I/O — General MC-344
Terminal I/O — ACCA Return Codes MC-345
Terminal I/O — Interprocessor Communications Return Codes MC-345
Terminal I/O — Virtual Terminal Communications Return Codes MC-345
Terminal I/O — 4975 Printer Return Codes MC-346
TP (Host Communication Facility) Return Codes MC-347
Unmapped Storage Return Codes MC-349

X.21 Circuit Switched Network Return Codes MC-350
Return Codes Listed by Number MC-351

Chapter 6. Stop Codes MC-375

Appendix A. Program Checks and Processor Status Word (PSW) MC-379

How to Interpret the Program Check Message MC-379

Format 1: Standard Program Check Messages MC-379

Processor Status Word MC-381

Processor Status Word Bit Descriptions MC-382

Format 2: Program Check Message from \$EDXIT MC-384

Glossary of Terms and Abbreviations MC-387

Index MC-397

Part 1. Messages

Part 1 contains many of the messages issued by the following utilities, modules, licensed programs, and system functions:

- \$BSCTRCE
- \$BSCUT1
- \$BSCUT2
- \$COMPRES
- \$COPY
- \$COPYUT1
- \$DASDI
- \$DEBUG
- \$DICOMP
- \$DIINTR
- \$DIUTIL
- \$DISKUT1
- \$DISKUT2
- \$DIRECT
- \$DUMP
- \$EDXASM
- \$EDXLINK
- \$GPIBUT1
- \$HXUT1
- \$IAM (*)
- \$IAMUT1 (*)
- \$IMAGE
- \$INITDSK
- IOLOADER
- \$IOTEST
- \$JOBQUT

-
- \$JOBUTIL
 - \$LOG
 - \$MSGUT1
 - \$MTM(*)
 - \$PREFIND
 - \$RJE2780,\$RJE3780
 - \$RJESNA (*)
 - \$RMU
 - \$\$SNA (*)
 - \$\$S1ASM
 - \$\$STGUT1
 - \$\$SUBMIT
 - \$\$S1S1UT1
 - \$TAPEUT1
 - \$TERMUT1
 - \$TERMUT2
 - \$TRAP
 - \$UPDATE
 - \$UPDATEH
 - \$VERIFY (*)
 - \$XPSLINK
 - \$XPSPOST
 - \$XPSPRE
 - Loader
 - Printer Spooling
 - Sensor I/O (*)
 - Sort/Merge (*)

An asterisk (*) indicates a licensed program or a utility related to a licensed program. A licensed program is a separately purchased program which is not part of the basic Event Driven Executive system.

Part 1 consists of the following chapters:

- Chapter 1, "Introduction to Messages" explains how the messages are organized and documented. It also gives tips for locating hard-to-find messages.
- Chapter 2, "Messages" contains messages arranged in alphabetical order.

Chapter 1. Introduction to Messages

The messages in Chapter 2 are generated during:

- System and device initialization
- Execution of a utility
- Execution of an application program
- Execution of a licensed program
- Assembling a source program
- Compiling a source program

Most of the messages in this book indicate abnormal or error conditions. Others provide information during normal operations. Most messages require you to take some action, either to restore conditions to normal, or to allow normal operations to proceed.

Self-explanatory information messages that do not require any user response, as well as most prompts, are not included in this book.

Introduction to Messages

How Messages are Documented

The documentation for each message includes the following information:

Message Text: Shows the actual message (in boldface type) worded exactly as issued. Variables are indicated by lower case letters.

Issued By: Identifies the utility, program product, or system function that issued the message.

Explanation: Explains the conditions or actions that caused the message to be issued.

System Action: States what the utility or system function does in response to the condition. When there is no action taken, "None" appears.

User Response: Explains what action you must take to resume normal operations.

Identifying Variables in Messages

If you get a message that contains the name of a particular data set, volume, terminal, or other device, that name is a *variable*. Also, if you get a message that contains a number, such as an address, storage size, record number, or buffer size, that number is probably a *variable*.

Variables are represented in this book as follows:

xxx = First or only variable in a message
yyy = 2nd variable in a message
zzz = 3rd variable in a message
nnn = 4th variable in a message

Note that they consist of three lowercase letters. In this book, representations of variables can stand for any combination of letters or numbers, regardless of the number and type of characters that actually appear in a variable in a message.

For example, you may get the following message:

DATA SET SAMPLE1,VOLUME1 NOT FOUND

You know that SAMPLE1,VOLUME1 are variables because they refer to one of your own data sets. In the book, you would find the message documented as follows:

DATA SET xxx,yyy NOT FOUND

Finding Messages in the Book

The messages are in alphabetical order, but here are a few tips to help you locate some that might seem hard to find.

| How Messages are Alphabetized

The messages in this book are alphabetized letter-by-letter rather than word-by-word. The sample list of messages that follows illustrates the order in which messages appear:

A COPY STATEMENT IS NOT ALLOWED WITH COPY CODE

ACTION= PARAMETER IS INCORRECT

ACTIVE STRUCTURE IS NOT DO

ACTIVE STRUCTURE IS NOT IF

ADAPTER TYPE NOT SUPPORTED

A DATA ADDRESS MUST BE SPECIFIED

ADDRESS= MUST BE SPECIFIED

Notice, for example, that the message **ADAPTER TYPE NOT SUPPORTED** appears before the message **A DATA ADDRESS MUST BE SPECIFIED**. The fourth letter in the word **ADAPTER** is a “P” while the fourth letter in the message **A DATA ADDRESS MUST BE SPECIFIED** is a “T”. The messages beginning with the word A are not grouped together, but are ordered according to the specific letters which make up the message.

Finding Messages that Begin with Special Characters

Some messages begin with *special characters*: the pound sign (#), the dollar sign (\$), and the asterisk (*). To find such a message, look at the letter that follows the special character. The message is listed alphabetically according to this letter.

Introduction to Messages

Finding Messages in the Book (*continued*)

Here is an example of finding a message that begins with a special character. Assume that the following message appears on your terminal:

```
$DISKUT3 LOAD ERROR ENCOUNTERED
```

You know that “\$” is a special character, so you note the first letter that comes after it in the message, which is “D”. Going to Chapter 2, you look for the message under the listings for “D”. You find the message documented the same way it was displayed:

```
$DISKUT3 LOAD ERROR ENCOUNTERED
```

Finding Messages that Begin with Numbers

A few messages begin with numbers that refer to things such as index registers and printers. These messages are at the end of the list of messages. For example, you might receive this message:

```
4975-01L REQUIRES BIT RATE OF 2400
```

Since the message begins with a number, you would go to the end of Chapter 2 to find it.

Other Hints for Finding Messages

You may get a message that can be issued by several different utilities. The message texts may be the same, but the explanations are probably different. Make sure you check the heading “Issued by” when you look up a message. Then you will know that the message explanation you are reading is the right one.

Chapter 2. Messages

Tips for Finding Messages in This Chapter

To find a message that begins with a special character, (#, \$, *) look for it under the first letter that follows the character.

To find a message that begins with a number, go to the end of Chapter 2.

Some messages are issued by more than one utility. Check the “Issued by” part of the message, to be sure that you are reading the correct explanation for the message you received.

A

A COPY OF THE NEW DIRECTORY EXISTS ON \$WRKDIR,IPL VOLUME

Issued by: \$IAMUT1

Explanation: During the process of renaming the new directory data set to \$IAMDIR (using the AL or AS subcommands of DR), an error occurred. The new directory data set name is \$WRKDIR residing on the IPL volume.

System Action: Ends the operation.

User Response: Use \$DISKUT1 to rename \$WRKDIR to \$IAMDIR.

A COPY STATEMENT IS NOT ALLOWED WITHIN COPY CODE

Issued by: \$EDXASM

Explanation: The copy code source module contains another COPY statement. Copy code cannot contain references to other copy code.

System Action: The error is flagged and compilation continues.

User Response: Remove the nested COPY statement from the source module by rearranging code or by converting copy code to in-line code. Recompile the program.

ACTION= PARAMETER IS INCORRECT

Issued by: \$EDXASM

Explanation: The parameter for the ACTION= operand of the GETEDIT/PUTEDIT instruction is specified incorrectly. Valid parameters are "IO" and "STG".

System Action: The error is flagged and compilation continues.

User Response: Specify either ACTION=IO or ACTION=STG.

ACTIVE STRUCTURE IS NOT DO

Issued by: \$EDXASM

Explanation: The active structure is not DO or there was no active structure.

System Action: The error is flagged and compilation continues.

User Response: Ensure the DO logic is correct.

ACTIVE STRUCTURE IS NOT IF

Issued by: \$EDXASM

Explanation: \$EDXASM has encountered instructions that are part of an "IF" structure (IF, ELSE or ENDIF), but the structure being processed is not an "IF" structure. This error can occur when the instruction terminating the previous structure was not recognized or was omitted.

System Action: The error is flagged and compilation continues.

User Response: Check the structure preceding the "IF" structure and code a valid terminating instruction for it.

**ADAPTER=MFA IS INVALID WITH X.21
TYPE=AC OR TYPE=DC**

Issued by: \$EDXASM

Explanation: You coded ADAPTER=MFA on the BSCLINE statement when you specified "AC" (auto-call) or "DC" (direct-call) for the TYPE operand. The ADAPTER operand is not valid with TYPE=AC or TYPE=DC.

System Action: The error is flagged and compilation continues.

User Response: Remove the ADAPTER operand from the BSCLINE statement.

**ADAPTER OR DEVICE ADDRESS, OR
DEVICE NAME INCORRECTLY
SPECIFIED**

Issued by: \$EDXASM

Explanation: The ADDRESS= operand on the ADAPTER or TERMINAL statement must be specified for MFA and ALPA adapters. The device address must be within the address range of the adapter. In addition, the device name specified must be one of the valid device types supported.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid address and device name.

**ADAPTER= PARAMETER IS INCORRECTLY
SPECIFIED**

Issued by: \$EDXASM

Explanation: The ADAPTER operand is coded incorrectly on the BSCLINE statement, or the label is missing on the BSCLINE statement where ADAPTER= is coded correctly.

System Action: The error is flagged and compilation continues.

User Response: Ensure the BSCLINE statement has a label and the ADAPTER= operand is coded correctly.

ADAPTER STATEMENT FOLLOWS END=YES ADAPTER STATEMENT

Issued by: \$EDXASM

Explanation: All ADAPTER statements must be grouped together, preceding the definition of the first device attached to an MFA or ALPA adapter. The last ADAPTER statement must have END=YES specified.

System Action: The error is flagged and compilation continues.

User Response: Group all ADAPTER statements together with the last statement specifying END=YES.

ADAPTER TYPE NOT SUPPORTED

Issued by: \$EDXASM

Explanation: The type specified on the ADAPTER= operand is invalid for this device.

System Action: The error is flagged and compilation continues.

User Response: Read the description of the ADAPTER operand for the device types that can be specified for this adapter.

A DATA ADDRESS MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The address of a 1-word storage area for the queue entry must be specified on the FIRSTQ, LASTQ, and NEXTQ instructions.

System Action: The error is flagged and compilation continues.

User Response: Specify the address or label of the area for "loc" on the FIRSTQ, LASTQ, and NEXTQ instructions.

A DATA SET MUST PRECEDE A COMMA

Issued by: \$SUBMIT

Explanation: You left out the name of the data set in response to a prompt. The format of your response should be a data set name, a comma, and a volume (dataset,volume).

System Action: Returns to command mode.

User Response: Enter a valid data set name followed by a comma and a volume name. Do not put a space between the data set name and the comma.

ADDRESS= MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The ADDRESS= operand is required and the address specified must be in the range of X'00' to X'FF'.

System Action: The error is flagged and compilation continues.

User Response: Specify the ADDRESS= operand in the range of X'00' to X'FF'

ADDR FIELD TOO LARGE - CAxxx TERMINATED

Issued by: \$S1ASM

Explanation: The device address operand on the CAxxx instruction contains a value greater than 255.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a two-digit hexadecimal device address operand that is less than or equal to 225, and reassemble.

ADDR FIELD TOO LARGE - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The device address operand on the previous CAxxx instruction contains a value greater than 255.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a two-digit hexadecimal device address operand that is less than or equal to 255, and reassemble.

ADDR FIELD TOO SMALL - CAxxx TERMINATED

Issued by: \$S1ASM

Explanation: The device address operand on the CAxxx instruction contains a value less than zero.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a two-digit hexadecimal device address operand that is greater than or equal to zero, and reassemble.

ADDR FIELD TOO SMALL - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The device address operand on the previous CAxxx instruction contains a value less than zero.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a two-digit hexadecimal device address operand that is greater than or equal to zero, and reassemble.

ADDR WAS NOT SUPPLIED-CAxxx TERMINATED

Issued by: \$S1ASM

Explanation: The device address operand on the CAxxx instruction is a required operand and was not supplied.

System Action: Terminates the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid device address operand, and reassemble.

ADDR SPECIFIED INCORRECTLY - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The device address operand on the previous CAxxx instruction has an invalid hexadecimal entry in the first and/or second digits.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a valid device address operand whose first and second digits are valid hexadecimal entries, and reassemble.

ADDR WAS NOT SUPPLIED - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The device address operand on the previous CAxxx instruction is a required operand and was not supplied.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a valid device address operand, and reassemble.

A DEVICE ADDRESS MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The ADDRESS= operand is required and must be within the range X'00' to X'FF'.

System Action: The error is flagged and compilation continues.

User Response: Specify the address within X'00' to X'FF'.

ALL DIRECTORY ENTRIES WILL BE DELETED, CONTINUE (Y/N)?

Issued by: \$IAMUT1

Explanation: On a request to reallocate the directory data set (using the AL or BN subcommands of DR), this prompt reminds you that all entries will be deleted, and requests confirmation to reallocate.

System Action: Awaits operator response.

User Response: Reply "Y" to delete all entries and reallocate the directory data set. Reply "N" to receive another prompt for the type of allocate request desired.

ALLOCATE ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: During the process of allocating the directory data set (using the AL subcommand of DR) an error occurred. The result was a \$DISKUT3 return code of xxx.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

ALLOCATE FAILED FOR DATASET - RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the DF command, an attempt to allocate the indexed file resulted in a \$DISKUT3 return code of xxx.

System Action: Current function terminated.

User Response: Examine the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

ALLOCATION CONSISTENCY ERROR

Issued by: \$HXUT1

Explanation: \$HXUT1 encountered a data set label that is inconsistent with the preceding labels on this diskette. For example, the areas defined for two data sets overlap. One of three possible messages immediately follows this message identifying the specific consistency error.

System Action: None.

User Response: Refer to the message you received indicating the specific consistency error to correct the problem.

ALTERNATE SECTOR IN ERROR AFTER ASSIGN - RETRY

Issued by: \$DASDI

Explanation: In attempting to verify the data after the alternate assignment was complete, an error was detected in the assigned alternate.

System Action: Processing continues.

User Response: Assign another alternate sector for the alternate just assigned.

AN IF/ELSE/ENDIF OR DO/ENDDO CLAUSE HAS NOT BEEN TERMINATED

Issued by: \$EDXASM

Explanation: When the ENDPROG statement was encountered, it was determined an IF/ELSE/ENDIF or DO/ENDDO structure was active.

System Action: The error is flagged and compilation continues.

User Response: Ensure that all coding structures are terminated with the appropriate ending statement.

AN INVALID KEYWORD PARAMETER WAS SPECIFIED

Issued by: \$EDXASM

Explanation: A parameter specified for a keyword operand was not valid for the instruction.

System Action: The error is flagged and compilation continues.

User Response: Determine the valid parameter values for the keyword operands in the instruction and code them accordingly.

ANY MORE DATA TO BE LOADED?

Issued by: \$IAMUT1

Explanation: Using the LO command, an end of data condition was encountered on the input data set. It is possible that more input data exists in another data set.

System Action: Awaits operator response.

User Response: Reply "Y" to receive a prompt for the input data set. Reply "N" to end the function normally.

**ASSOCIATED PRIMARY ENTRY
(DSNAME,VOLUME)**

Issued by: \$IAMUT1

Explanation: Using the IE subcommand of DR, you are prompted for the name of the primary data set and volume associated with the new secondary entry.

System Action: Awaits operator response.

User Response: Enter the data set name and volume of the primary indexed file. If you omit the volume, the IPL volume is assumed.

**ASSOCIATED SECONDARY ENTRIES WILL
BE DELETED, CONTINUE (Y/N)?**

Issued by: \$IAMUT1

Explanation: The directory entry to be deleted (using the DE subcommand of DR) is a primary entry with associated secondary entries. This prompt reminds you that the secondary entries will also be deleted, and requests confirmation of the deletes.

System Action: Awaits operator response.

User Response: Reply "Y" to delete the primary entry and all its secondaries. Reply "N" to cancel the delete.

**ATTACHMENT RPQ D02242 ACTIVE
ATTACHMENT RPQ D02241 INACTIVE**

Issued by: \$S1S1UT1

Explanation: The primary processor is not active. The secondary processor is active. The attachment is not complete.

System Action: Returns to command mode.

User Response: Wait and retry. If possible, contact the operator of the primary processor to have it turned on.

ATTACHMENT RPQ D02241 ACTIVE ATTACHMENT RPQ D02242 INACTIVE

Issued by: \$S1S1UT1

Explanation: The primary processor is active. The secondary processor is not active. The attachment is not complete.

System Action: Returns to command mode.

User Response: Wait and retry. If possible, contact the operator of the secondary processor to have it turned on.

ATTNLIST: SCOPE= MUST BE LOCAL OR GLOBAL

Issued by: \$EDXASM

Explanation: The only valid operands for an ATTNLIST instruction are "LOCAL" or "GLOBAL".

System Action: The error is flagged and compilation continues.

User Response: Ensure the local or global operand is spelled correctly.

ATTEMPT TO READ MORE RECORDS WITH 1 STATEMENT THAN ARE AVAILABLE

Issued by: \$TAPEUT1

Explanation: You attempted to read more records than exist in the file.

System Action: Issues a tape read/write return code and continues processing.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

ATTNLIST COMMAND NAME MUST BE 1-8 CHARACTERS IN LENGTH

Issued by: \$EDXASM

Explanation: Entries in the ATTNLIST command name must not exceed 1–8 characters in length.

System Action: The error is flagged and compilation continues.

User Response: Ensure the command string does not exceed eight characters in length.

BITS= INVALID FOR AI AND AO

Issued by: \$EDXASM

Explanation: The BITS= operand cannot be specified for AI, AO, DO PULSE, or external sync DI/DO operations.

System Action: The error is flagged and compilation continues.

User Response: Do not code the BITS= operand for the sensor I/O operations previously described.

**BITS= MUST BE SPECIFIED WITH
TYPE=SUBGROUP**

Issued by: \$EDXASM

Explanation: The BITS= operand must be specified if TYPE=SUBGROUP is coded on the IODEF statement.

System Action: The error is flagged and compilation continues.

User Response: Specify the starting bit and bit length for the BITS= operand.

BLOCK OUT OF SEQUENCE. RBN xxx

Issued by: \$VERIFY

Explanation: A data block which is used but empty nevertheless has a high key assigned to it. In this case, the block is out of sequence according to that high key.

System Action: Verification continues.

User Response: Rebuild the \$IAM file.

**# BLOCKS IN FREEPOOL CHAIN DOES
NOT MATCH FREE POOL COUNT IN
FCB**

Issued by: \$VERIFY

Explanation: The FCB contains a count of the number of blocks in the freepool chain. That count does not match the actual number of blocks as counted by \$VERIFY.

System Action: Verification continues.

User Response: Rebuild the \$IAM file.

BLOCKSIZE MUST BE NUMERIC

Issued by: \$IAMUT1

Explanation: Using the LO or UN commands, the response to the block size must be a decimal value up to 4 digits.

System Action: The block size prompt is reissued.

User Response: Enter a decimal value up to 4 digits.

**BOE GREATER THAN EOE FOR DATA SET
xxx DEFINED AT RECORD yyy
DATA SET NOT CONSIDERED
ALLOCATED**

Issued by: \$HXUT1

Explanation: This message is preceded by "ALLOCATION CONSISTENCY ERROR". The beginning of extent (BOE) is larger than the end of extent (EOE) in the H-exchange diskette.

System Action: None.

User Response: Enter the UD command (update data set label by name) or the UH command (update data set label by number) and specify the correct starting and/or ending address (if known) to the prompt(s) for the beginning and ending extents.

**BOE OR EOE DEFINED IN HEADER LABEL
AREA FOR DATA SET xxx DEFINED AT
RECORD yyy DATA SET NOT CONSIDERED
ALLOCATED**

Issued by: \$HXUT1

Explanation: This message is preceded by "ALLOCATION CONSISTENCY ERROR". Either the beginning of extent (BOE) or the end of extent (EOE) of a data set points to an area in the H-exchange data set label area.

System Action: None.

User Response: Enter the UD command (update data set label by name) or the UH command (update data set label by number) and specify the correct starting and/or ending address (if known) to the prompt(s) for the beginning and ending extents.

**BRACKET ERROR ENCOUNTERED ON LINE
#xxx**

Issued by: \$MSGUT1

Explanation: There is a syntax error in the right set of parameter brackets.

System Action: Continues processing source data set.

User Response: When processing finishes, correct the error and retry.

BSF ERROR - TAPE RC = xxx

Issued by: \$TAPEUT1

Explanation: An I/O error occurred during the backspace file function. The result was tape read/write return code of xxx.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

BSR ERROR - TAPE RC = xxx

Issued by: \$TAPEUT1

Explanation: An I/O error occurred during the backspace record function. The result was a tape read/write return code of xxx.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

BUFF ADDRESS IS NUMERIC - CAWRITE TERMINATED

Issued by: \$S1ASM

Explanation: The BUFFER operand on the CAWRITE instruction is numeric. The first character must be alphabetic.

System Action: Terminates the CAWRITE instruction.

User Response: On the CAWRITE instruction, specify a valid BUFFER operand whose first character is alphabetic, and reassemble.

BUFFER DOES NOT CHECK OUT ABORT ECHO TEST

Issued by: \$S1S1UT1

Explanation: The echo test sends 1024 bytes of data back and forth to the CPU. The data does not contain the correct values.

System Action: Terminates function.

User Response: Retry the function.

BUFFER LENGTH EXCEEDS 16380 WORDS

Issued by: \$EDXASM

Explanation: The length specified on the BUFFER statement exceeded 16380 words. The next operand determines whether this length is in bytes or words.

System Action: The error is flagged and compilation continues.

User Response: Specify a buffer length not exceeding 16380 words (32760 bytes).

BUFFER MAY NOT BE INDEXED - CAIOCB TERMINATED

Issued by: \$EDXASM

Explanation: The BUFFER= operand on the CAIOCB statement is specified as an indexable address.

System Action: Terminates the CAIOCB expansion.

User Response: On the CAIOCB statement, specify the BUFFER= operand as a nonindexable address, and reassemble.

BUFFER MUST BE AN ADDRESS - CAIOCB TERMINATED

Issued by: \$S1ASM

Explanation: The BUFFER= operand on the CAIOCB statement is not supplied as an address.

System Action: Terminates the CAIOCB expansion.

User Response: On the CAIOCB statement, specify the BUFFER= operand as an address, and reassemble.

BUFFER POOL EXHAUSTED

Issued by: \$BSCUT2

Explanation: An internal error was detected. The system buffer overflowed.

System Action: Terminates function.

User Response: Check the system output to find the error; correct and retry.

B

BUFFER SIZE xxx

Issued by: \$IAMUT1

Explanation: Using the BF command, this message displays the current size of the Indexed Access Method buffer and provides you the opportunity to reset the size.

System Action: If you respond with a null input, no action is taken. If you respond with a new size, the new size is stored and then is written to disk as the new buffer size.

User Response: Reply with a null input to allow the buffer size to remain as it is. Reply with a new buffer size if you desire to change it.

BUFFER SMALLER THAN INPUT OR OUTPUT RECORD. INCREASE BUFFER SIZE BEFORE REATTEMPTING OPERATION

Issued by: \$IAMUT1

Explanation: There is insufficient program storage to contain an input or output record.

System Action: Terminates command.

User Response: Increase the dynamic program size using the SS command of the \$DISKUT2 utility. The buffer size must be large enough to contain the entire input and/or output record (whichever is larger).

BUFFER SUPPLIED HAS MORE THAN 8 CHARACTERS

Issued by: \$S1ASM

Explanation: The BUFFER= operand on the CAIOCB statement has more than eight characters.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a BUFFER= operand that is less than or equal to eight characters in length, and reassemble.

BUFFER WAS NOT SUPPLIED - CAIOCB TERMINATED

Issued by: \$EDXASM, \$S1ASM

Explanation: The BUFFER= operand on the CAIOCB statement is required and was not supplied.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a valid BUFFER= operand, and reassemble.

**BUFF NAME IS NUMERIC - EXPANSION
TERMINATED**

Issued by: \$S1ASM

Explanation: The BUFFER operand on the CAWRITE instruction is numeric. The first character must be alphabetic.

System Action: Terminates the expansion.

User Response: On the CAWRITE instruction, specify a valid BUFFER operand whose first character is alphabetic, and reassemble.

**BUFF NOT SPECIFIED - EXPANSION
TERMINATED**

Issued by: \$EDXASM

Explanation: The BUFFER operand on the CAWRITE instruction is a required operand and is not specified.

System Action: Terminates the expansion.

User Response: On the CAWRITE instruction, specify a valid BUFFER operand, and reassemble.

**BUFF NAME TOO LONG - CAWRITE
TERMINATED**

Issued by: \$S1ASM

Explanation: The BUFFER operand on the CAWRITE instruction has more than eight characters.

System Action: Terminates expansion of the CAWRITE instruction.

User Response: On the CAWRITE instruction, specify a BUFFER operand that is less than or equal to eight characters in length, and reassemble.

**CAIOCB NAME IS NUMERIC - CAxxx
TERMINATED**

Issued by: \$S1ASM

Explanation: The CAIOCB operand on the previous CAxxx instruction is numeric. The first character must be alphabetic.

System Action: Terminates the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid CAIOCB operand whose first character is alphabetic, and reassemble.

**CAIOCB NAME TOO LONG - CAxxx
TERMINATED**

Issued by: \$S1ASM

Explanation: The CAIOCB operand on the CAxxx instruction has more than eight characters.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a CAIOCB operand that is less than or equal to eight characters in length, and reassemble.

**CAIOCB NAME IS NUMERIC - EXPANSION
TERMINATED**

Issued by: \$EDXASM

Explanation: The CAIOCB operand on the CAxxx instruction is numeric. The first character must be alphabetic.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid CAIOCB operand whose first character is alphabetic, and reassemble.

**CAIOCB NOT SPECIFIED - EXPANSION
TERMINATED**

Issued by: \$EDXASM

Explanation: The CAIOCB operand on the previous CAxxx instruction is a required operand and is not specified.

System Action: Terminates the CAxxx expansion.

User Response: On the CAxxx instruction, specify a valid CAIOCB operand, and reassemble.

**CANNOT ALLOCATE DATA SET xxx
RC= yyy**

Issued by: Session Manager

Explanation: While trying to submit a background job, the session manager tried unsuccessfully to allocate a temporary \$JOBQ file on EDX003.

System Action: The session manager does not submit the job. It returns to the previous menu.

User Response: Check the \$DISKUT3 return codes for the cause of the error and take the appropriate action.

CANNOT LOAD \$JOBQ

Issued by: Session Manager

Explanation: You tried to submit a background job without first loading the job queue processor, \$JOBQ. The session manager tried to load \$JOBQ for you, but not enough free storage is available in the system.

System Action: The session manager does not submit the job. It returns to the previous menu.

User Response: Ensure that one of the partitions has approximately 32K bytes of free storage. Resubmit your background job.

**CANNOT REASSIGN ADDRESS OF
LOADING TERMINAL**

Issued by: \$TERMUT1

Explanation: You attempted to change the address of the terminal you are currently using. The supervisor does not allow this.

System Action: Returns to command mode.

User Response: Either reassign a different terminal or go to another terminal and reassign your original terminal.

CAPS= INVALID WITH IMMEDIATE TEXT

Issued by: \$EDXASM

Explanation: If the "loc" operand is specified on the READTEXT as an immediate text character string, the CAPS operand is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code either the CAPS operand or an immediate text character string but not both.

CAPS= INVALID WITH XLATE=NO

Issued by: \$EDXASM

Explanation: The CAPS= operand cannot be specified on the READTEXT or PRINTTEXT instructions if XLATE=NO is specified.

System Action: The error is flagged and compilation continues.

User Response: Specify either the CAPS= operand or XLATE=NO but not both.

**CAPS MUST BE SPECIFIED AS
CAPS=Y(ES) OR A POSITIVE INTEGER**

Issued by: \$EDXASM

Explanation: The CAPS= operand for folding to upper case characters on the READTEXT or PRINTTEXT must be specified as either CAPS=Y or CAPS=YES or CAPS= character count.

System Action: The error is flagged and compilation continues.

User Response: Correct the error as appropriate and recompile.

**CDMINIT2 WAITING FOR STORAGE TO
BECOME AVAILABLE FOR COMMAND
PROCESSOR**

Issued by: \$MTM

Explanation: CDMINIT2 is waiting to complete Communications Facility (CF) initialization because communications facility buffer has insufficient storage to start MTM/CF station.

System Action: Initialization waits until sufficient storage is available.

User Response: Delete messages using Communications Facility utility \$UT1.

CHECKSUM ERROR

Issued by: \$\$S1S1UT1

Explanation: During transmissions between the two processors, the check sums indicate that data either not sent or not received correctly.

System Action: Terminates function.

User Response: Retry the function.

**CLOSE ERROR. RC = xxx TAPE NOT
INITIALIZED**

Issued by: \$TAPEUT1

Explanation: An error occurred while attempting to initialize a tape. The result was a tape read/write return code of xxx.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

CLOSE ERROR - TAPE RC = xxx

Issued by: \$TAPEUT1

Explanation: An error occurred while attempting to close and set the tape offline. The result was a tape read/write return code of xxx.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

CODE MUST BE SELF-DEFINING TERM

Issued by: \$EDXASM

Explanation: The CODE= operand on the DEQ instruction must be explicitly defined as a positive or negative integer.

System Action: The error is flagged and compilation continues.

User Response: Specify an explicit positive or negative integer for CODE=.

COMMAND= MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The COMMAND= operand of the IDCB instruction is required and was not specified.

System Action: The error is flagged and compilation continues.

User Response: Code the IDCB instruction with the COMMAND= operand and valid parameters.

COMMAND PROCESSOR BUSY

Issued by: \$RJESNA

Explanation: You tried to enter an \$RJESNA operator command before \$RJESNA was finished processing the previously entered command.

System Action: \$RJESNA continues to process the previous command.

User Response: Wait and retry the command.

COMMUNICATIONS FACILITY NO LONGER ACTIVE

Issued by: \$MTM

Explanation: Communications Facility was active during the execution of the CDMINIT2 initialization routine, but CDMINIT2 then detected that the Communications Facility was no longer active. Some stations may have been started and some asynchronous tasks may have been loaded.

System Action: The initialization process terminates after this message is issued. Multiple Terminal Manager will be able to operate with all non-Communications Facility functions.

User Response: Terminate Multiple Terminal Manager, restart Communications Facility, and then restart Multiple Terminal Manager.

COMP= MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: COMP= operand was not specified, and it is required.

System Action: The error is flagged and compilation continues.

User Response: Specify the COMP= operand on this instruction.

COMPID MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: COMPID operand was not specified for the COMP instruction, and it is a required operand.

System Action: The error is flagged and compilation continues.

User Response: Specify COMPID on the COMP instruction as a 1-4 character ID enclosed in quotes.

COMPID MUST BE ENCLOSED IN QUOTES

Issued by: \$EDXASM

Explanation: COMPID operand on the COMP instruction was not enclosed in quotes.

System Action: The error is flagged and compilation continues.

User Response: Specify COMPID on the COMP instruction as a 1-4 character ID enclosed in quotes for the message data set or storage resident message area.

COMPID MUST BE 1-4 CHARACTERS IN LENGTH

Issued by: \$EDXASM

Explanation: COMPID specified was not 1-4 characters in length.

System Action: The error is flagged and compilation continues.

User Response: Specify COMPID on the COMP instruction as a 1-4 character ID enclosed in quotes for the message data set or storage resident message area.

CONCATENATION FILE xxx.yyy NOT FOUND

Issued by: \$RJE2780, \$RJE3780

Explanation: The file name and volume you specified for concatenation does not exist.

System Action: Ends the operation.

User Response: Check that the name you specify refers to an existing file.

CONDITION MUST BE EQ, NE, GT, LT, GE OR LE FOR STRING COMPARE

Issued by: \$EDXASM

Explanation: You did not specify one of the valid operators for a string compare operation.

System Action: The error is flagged and compilation continues.

User Response: Ensure one of the valid operators is specified for a string compare.

**CONSOLE NAME IS NUMERIC - \$\$SYSPRTR
BEING USED**

Issued by: \$S1ASM

Explanation: The CONSOLE= operand on the CAPRINT instruction is numeric. The first character must be alphabetic.

System Action: Uses the default console \$\$SYSPRTR.

User Response: If \$\$SYSPRTR is not the needed console, then on the CAPRINT instruction specify a valid CONSOLE= operand whose first character is alphabetic, and reassemble.

**CONSOLE= NAME TOO LONG - \$\$SYSPRTR
BEING USED**

Issued by: \$S1ASM

Explanation: The CONSOLE= operand on the CAPRINT instruction has more than eight characters.

System Action: Uses the default console \$\$SYSPRTR.

User Response: If \$\$SYSPRTR is not the needed console, then on the CAPRINT instruction specify a CONSOLE= operand that is less than or equal to eight characters long, and reassemble.

**CONSOLE NOT SPECIFIED - \$\$SYSPRTR
BEING USED**

Issued by: \$S1ASM

Explanation: The CONSOLE= operand on the CAPRINT instruction is required and is not specified.

System Action: Uses the default console \$\$SYSPRTR.

User Response: If \$\$SYSPRTR is not the needed console, then on the CAPRINT instruction you must specify the CONSOLE= operand, and reassemble.

**CONSOLE SPECIFIED INCORRECTLY -
DEFAULT ASSUMED**

Issued by: \$EDXASM

Explanation: The CONSOLE= operand on the CAPRINT instruction is incorrectly specified.

System Action: Uses the default console \$\$SYSPRTR.

User Response: If \$\$SYSPRTR is not the needed console, then on the CAPRINT instruction specify a valid CONSOLE= operand, and reassemble.

**COPY CODE ERROR - EXPANSION
TERMINATED**

Issued by: \$EDXASM

Explanation: The copy code required for CACB1 is not correct.

System Action: Terminates the expansion.

User Response: Verify that the copy code required is available in your library and is correct.

COPY CODE MODULE NOT DEFINED

Issued by: \$EDXASM

Explanation: A copy code statement was encountered in the source module, but the requested copy code module was not found in any of the copy code volumes.

System Action: The error is flagged and compilation continues.

User Response: You can:

1. Ensure that the name of the copy code module is coded correctly.
2. Edit the language control data set so that it contains the correct volumes.
3. Move the desired copy code module into an existing copy code volume.

**COUNT IS NOT ALLOWED WITH INDEX
REGISTERS**

Issued by: \$EDXASM

Explanation: Index registers and the count operand cannot be used in the same instruction (mutually exclusive).

System Action: The error is flagged and compilation continues.

User Response: Remove the count specification from the instruction and recompile.

**COUNT MAY NOT BE MORE THAN 2 WITH
REGISTER OPERANDS**

Issued by: \$EDXASM

Explanation: The "count" must not exceed 2 (words) for register move operations.

System Action: The error is flagged and compilation continues.

User Response: Ensure that "count" does not exceed 2 words.

**COUNT MUST BE A POSITIVE
SELF-DEFINING TERM**

Issued by: \$EDXASM

Explanation: Self-defining terms are decimal constants, hexadecimal constants, EBCDIC constants, or symbols preceded by a “+” or “-” sign. The “count” must be specified explicitly and not as an address or through a register.

System Action: The error is flagged and compilation continues.

User Response: Ensure the “count” is specified as a positive self-defining term.

COUNT NOT BETWEEN 0 AND 32767

Issued by: \$EDXASM

Explanation: The “length” or “count” on the TP instruction is too large or is negative. While the count on most TP instructions cannot exceed 256, the range allowed by \$EDXASM is 0–32767.

System Action: The error is flagged and compilation continues.

User Response: Specify the length or count between the range of 0–32767.

COUNT TOO HIGH IN PARM= OPERAND

Issued by: \$EDXASM

Explanation: The word count specified by the PARM= operand caused the program header to exceed the maximum size.

System Action: The error is flagged and compilation continues.

User Response: Refer to the *Language Reference* for usage of the PARM= operand of the PROGRAM instruction.

CRDELAY SPECIFIED INCORRECTLY

Issued by: \$EDXASM

Explanation: The carriage return delay (CRDELAY) operand of the TERMINAL statement is coded incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Examine and recode the CRDELAY operand of the TERMINAL statement.

**CSS ONLY REFLECTS LAST HARDWARE
ERROR; NOT SOFTWARE**

Issued by: \$S1S1UT1

Explanation: The cycle steal status may not reflect the most recent operation.

System Action: None.

User Response: This is an informational message. No user response is needed.

D

DATA ERR - ECC INVERTED

Issued by: \$DASDI

Explanation: One or more RBAs have an inverted ECC. This indicates than an unrecoverable data error occurred while moving the data in an alternate assignment operation. The integrity of the RBAs has been compromised.

System Action: Processing continues.

User Response: Refresh the RBAs. If problems persist, then assign an alternate sector to the RBAs.

DATA ERROR

Issued by: \$DASDI

Explanation: One or more RBAs have returned a data check to the read data. This indicates that the integrity of the data in this RBA has been compromised.

System Action: Processing continues.

User Response: Refresh the RBAs. If problems persist, then assign an alternate sector to the RBAs.

DATA SET ALLOCATION OVERLAP FOR DATA SET xxx DEFINED AT RECORD yyy DATA SET NOT CONSIDERED ALLOCATED

Issued by: \$HXUT1

Explanation: This message is preceded by "ALLOCATION CONSISTENCY ERROR". The extent defined for an H-exchange data set overlaps with another extent.

System Action: None.

User Response: Enter the UD command (update data set label by name) or the UH command (update data set label by number) and specify the correct starting and/or ending address (if known) to the prompt(s) for the beginning and ending extents.

DATA SET DSxxx NOT FOUND

Issued by: \$PREFIND

Explanation: The data set name you specified does not exist on the volume in use.

System Action: Returns to command mode.

User Response: Correct the data set name and/or volume and retry the function.

**DATA SET HAS NO CHECK POINT DATA,
RESTART INVALID****Issued by:** \$SPLUT1**Explanation:** When the spool utility tried to read data saved from a previous spool session, it found that the data was unusable.**System Action:** Ends the spool operation.**User Response:** You can:

- Start spooling over again from the beginning. Then rerun the spool job.
- Restart spooling with another spool data set.

**DATA SET INTEGRITY MAY BE
COMPROMISED****Issued by:** \$DIUTIL**Explanation:** If data exists on the device in use, it may be lost in the next procedure.**System Action:** Awaits operator response.**User Response:** Cancel the operation if desired, or proceed.**DATA SET xxx IS NOT A DATA TYPE****Issued by:** \$PREFIND**Explanation:** The data set name you specified refers to a “program” type data set. It must be a “data” type data set.**System Action:** Returns to command mode.**User Response:** Specify a data set that is of the data type and retry function.**DATA SET xxx IS NOT A PROGRAM TYPE****Issued by:** \$PREFIND**Explanation:** The data set name you specified refers to a “data” type data set. It must be a “program” type data set.**System Action:** Returns to command mode.**User Response:** Specify a data set that is of the program type and retry the function.**DATA SET NAME \$\$ --- NOT PERMITTED
FOR COPYCODE****Issued by:** \$EDXASM**Explanation:** A copy code module cannot have \$\$ as a prefix; this naming convention is reserved for system use.**System Action:** The error is flagged and compilation continues.**User Response:** Use \$DISKUT1 utility to rename the data set. Do not use \$\$ as a prefix in the name.

DATA SET NOT ALLOCATED

Issued by: \$TAPEUT1

Explanation: An I/O error occurred during the allocation process.

System Action: Issues a tape read/write return code. Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

DATA SET xxx,yyy NOT FOUND

Issued by: \$PREFIND

Explanation: The data set you specified does not exist on the volume in use.

System Action: Returns to command mode.

User Response: Specify correct data set and volume names.

DATA SET NOT FOUND IN DIRECTORY

Issued by: \$COPY

Explanation: A basic exchange data set was not found in the directory.

System Action: Returns to command mode.

User Response: Reenter a valid data set name.

**DATA SET xxx,yyy TOO SMALL,
REALLOCATE LARGER**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The specified data set is too small to be used in the function you want to perform.

System Action: Ends the operation.

User Response: Reallocate the specified data set to be larger.

**DATA SET xxx TOO SMALL TO HOLD A
NUCLEUS MUST BE AT LEAST 257
RECORDS**

Issued by: \$UPDATE

Explanation: The data set you specified to contain a nucleus contains less than the required number of records.

System Action: Returns to command mode.

User Response: Allocate a larger data set of at least 257 records to hold the nucleus. Then retry the function.

**DATA SET USED FOR SORT HAS
INCORRECT FORMAT**

Issued by: \$DIRECT

Explanation: In the sort data set you created, you entered one or more lines the wrong way.

System Action: Returns to command mode.

User Response: Edit the sort data set to correct the errors and rerun the function.

DATA TYPE MUST BE WORD OR BYTE

Issued by: \$EDXASM

Explanation: The data type for the BUFFER statement is not "WORD" or "BYTE".

System Action: The error is flagged and compilation continues.

User Response: Specify either "WORD" or "BYTE" for the BUFFER statement.

DATA TYPE MUST BE WORD WITH REGISTER OPERANDS

Issued by: \$EDXASM

Explanation: When registers are specified as operands, word precision must also be specified.

System Action: The error is flagged and compilation continues.

User Response: Specify word precision when using registers as operands.

DATA TYPE SPECIFICATION IS NOT RECOGNIZED

Issued by: \$EDXASM

Explanation: An invalid data type was used in a DATA or DC statement.

System Action: The error is flagged and compilation continues.

User Response: Examine data statements for invalid or incorrectly coded data constants.

DCB ADDRESS MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The DCB= operand must be specified when COMMAND=START or COMMAND=SCSS is coded on the IDCB instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the label of a DCB statement on the DCB= operand.

DECOMPACTION REQUESTED - ROUTINE NOT AVAILABLE

Issued by: \$RJESNA

Explanation: The host attempted to send a compacted data stream or a compaction table, but no decompaction routine has been supplied.

System Action: None.

User Response: To correct the condition, perform one of the following:

- Supply a decompaction routine.
- Check the host definition of the workstation and verify that compaction has not been specified.
- Check the link of the decompaction routine installation and verify correct completion.

DEFINED PRINTER IS BUSY

Issued by: \$RJESNA

Explanation: In an attempt to use a printer defined to \$RJESNA with a PRINTON command or initialization, \$RJESNA determined the printer was already in use by another task. A subsequent message, "ENTER C(CONTINUE), T(TERMINATE), R(REDIRECT):" is issued to allow recovery from this error.

System Action: Waits for response.

User Response: Retry the operation by entering a "C", or cancel the PRINT command by entering a "T". If you enter a "T", the host is notified that the print function is unavailable, and a command to the host is required to start the printer. To redirect the output to a different printer, specify "R".

DELETE ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the AL, AS, or BN subcommands of DR, an error occurred during the process of deleting the old directory data set. The result was a \$DISKUT3 return code of xxx.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

DELETED SECTOR DETECTED FOR DATA SET xxx,yyy AT RECORD zzz

Issued by: \$HXUT1

Explanation: The utility detected a deleted sector on an H-exchange diskette during a copy operation (RX or WX commands).

System Action: The utility then asks whether you want to continue or stop the copy operation and waits for your response.

User Response: Enter "Y" to stop the copy operation. Enter "N" to continue; the deleted sector is not copied (skipped) and the copy continues.

DELETE OF ?? VOLUME NAME INVALID IN JOB MODE

Issued by: \$PREFIND

Explanation: You did not specify an actual volume name for use in job mode.

System Action: Terminates function.

User Response: Specify a valid volume name and rerun the procedure.

DENSITY= MUST BE EITHER 1600, 3200, OR DUAL (DEFAULT=1600)

Issued by: \$EDXASM

Explanation: An invalid density was specified for the TAPE instruction for the 4968 tape device.

System Action: The error is flagged and compilation continues.

User Response: Specify either 1600, 3200, or DUAL for DENSITY=. If omitted, DENSITY=1600 is the default.

DENSITY= MUST BE EITHER 800, 1600, OR DUAL (DEFAULT=1600)

Issued by: \$EDXASM

Explanation: An invalid density was specified for the TAPE instruction for the 4969 tape device.

System Action: The error is flagged and compilation continues.

User Response: Specify either 800, 1600, or DUAL for DENSITY=. If omitted, DENSITY=1600 is the default.

DEVADDR POSITIONAL PARAMETER MISSING

Issued by: \$EDXASM

Explanation: The “devaddr” operand of the EXOPEN or EXCLOSE instruction is required and must be specified as 2-hexadecimal digits.

System Action: The error is flagged and compilation continues.

User Response: Specify the valid device address for “devaddr” as 2 hexadecimal digits within the range X'00' to X'FF'.

DEVICE ADDRESS MUST BE FROM X'00' TO X'FF'.

Issued by: \$EDXASM

Explanation: The device address must be within the range of X'00' to X'FF'.

System Action: The error is flagged and compilation continues.

User Response: Code the device address within the range X'00' to X'FF'.

DEVICE ADDRESS NOT ASSIGNED

Issued by: \$TERMUT1

Explanation: The terminal control block could not be found for the specified device.

System Action: Returns to command mode.

User Response: Retry the function specifying a different terminal name.

D

DEVICE AT ADDRESS xxx UNUSABLE

Issued by: \$DASDI

Explanation: A 4962 or 4963 disk or a 4964 or 4965 diskette device is marked unusable and this status message is printed if I/O operations to the device fail. Some possible causes are:

- Unrecoverable hardware error detected on this device at IPL
- Diskette inserted incorrectly
- Device definition conflict at IPL
- Device not powered on
- Device not ready at IPL

System Action: Terminates the initialization.

User Response: Check for problems that may exist with the diskette. Correct any problem and restart the initialization.

DEVICE AT ADDRESS 22 UNUSABLE

Issued by: \$DASDI

Explanation: A 4966 is searched in slot 1 of magazine A and slot 1 of magazine B. If a diskette is found in any one of these slots, searching continues until an empty slot is found. If no diskettes in the 4966 diskette device can be accessed, the device is marked unusable and this status message is printed. Some possible causes are:

- Unrecoverable hardware error detected on this device at IPL
- Diskette inserted incorrectly
- Device definition conflict at IPL
- Device not powered on
- Device not ready at IPL

System Action: Terminates the initialization.

User Response: Insert a new diskette in the 4966 diskette device and restart the initialization.

DEVICE CANNOT SUPPORT DISKETTE 2D

Issued by: \$HXUT1

Explanation: The diskette was inserted in a diskette drive other than a 4965 or 4966, or you referred to a fixed disk.

System Action: The utility rejects the diskette and asks again for the volume name.

User Response: Insert the diskette into a 4965 or 4966 diskette drive and reenter the volume name.

DEVICE IS BUSY REQUIRED BY xxx

Issued by: IOLOADER

Explanation: The xxx group (where xxx is PI, DI, or DO) with external sync defined in the IODEF statement is in use by another resource.

System Action: The program load is terminated.

User Response: Check the IODEF or SENSORIO statement and ensure the group xxx is not in use, then retry.

DEVICE IS BUSY REQUIRED BY xxx

Issued by: \$IOTEST

Explanation: The device you tried to use is busy (in use by another program).

System Action: Returns to command mode.

User Response: Wait until the device is free and retry.

DEVICE IS NOT A S1S1 ATTACHMENT

Issued by: \$\$S1S1UT1

Explanation: The device address you specified does not refer to a Series/1 to Series/1 attachment device.

System Action: Terminates function.

User Response: Determine the supported attachment(s) for your system. Retry the function specifying a valid device address.

DEVICE IS NOT READY

Issued by: \$DASDI

Explanation: The device is not turned on, is not plugged in properly, or is for some other reason not operating.

System Action: Returns to command mode.

User Response: Check the device to see that it is turned on and operational.

DEVICE IS NOT SUPPORTED

Issued by: \$DASDI

Explanation: While trying to varyoff a device in preparation for initialization, the system found that it does not support the device.

System Action: Ends the operation.

User Response: This is an informational message. No user response is needed.

DEVICE NOT OFFLINE. DO YOU WISH TO CONTINUE?

Issued by: \$TAPEUT1

Explanation: The operation you are performing will automatically vary off the device. The device may be in use by another program.

System Action: Awaits operator response.

User Response: Check to see why the device is online. If it is in use, answer the prompt "N". If it is not in use, answer "Y".

D

DEVICE NOT SUPPORTED

Issued by: \$IOTEST

Explanation: You issued a command for a specific device type, and the device name you specified is not that type of device.

System Action: Returns to command mode.

User Response: Determine the proper device names with the LS command. Then retry the function.

DEVICE NOT SUPPORTED BY MFA

Issued by: \$EDXASM

Explanation: The device specified is not supported by the multifunction adapter (MFA).

System Action: The error is flagged and compilation continues.

User Response: Refer to the *Installation and System Generation Guide* for devices that are supported by the MFA.

DEVICE SPECIFIED IS NOT A SERIES/1 - SERIES/1

Issued by: \$S1S1UT1

Explanation: The address you specified does not refer to a S/1-S/1 device.

System Action: Terminates function.

User Response: Determine the addresses of S/1-S/1 device with the \$IOTEST utility. Retry the function specifying the valid device address.

DEVICE SPECIFIED NOT SUPPORTED FOR DEVICE SAVE

Issued by: \$TAPEUT1

Explanation: The "type" code contained in the DDB of the device to be saved is not supported.

System Action: Returns to command mode.

User Response: Enter another command if necessary.

DEVICE TYPE INVALID

Issued by: \$MTM

Explanation: The device type specified for the TERMINAL file record listed immediately before this message is invalid or TYPE=3101 and no 3101 support was included.

System Action: Does not connect the terminal.

User Response: Edit the TERMINAL record to reflect the correct terminal type or rebuild Multiple Terminal Manager with the 3101 support. Stop and restart the manager.

DEVICE TYPE IS NOT SPOOLABLE

Issued by: \$EDXASM

Explanation: A device was incorrectly specified as spoolable.

System Action: The error is flagged and compilation continues.

User Response: Remove the SPOOL operand from the TERMINAL statement.

DIRECTORY DATA SET NOT FOUND

Issued by: \$IAMUT1

Explanation: An attempt was made to open the directory data set, but the directory data set could not be found.

System Action: Terminates the operation.

User Response: Create the directory data set using the AL subcommand of DR.

**DIRECTORY ENTRY BEFORE xxx IS
INCORRECT COMPRESS ABORTED**

Issued by: \$COMPRES

Explanation: \$COMPRES was unable to continue because it found an error in the directory at the point specified by xxx. No compress occurred, but the directory was altered so that members are in order according to location on disk or diskette.

System Action: The \$COMPRES utility ends.

User Response: Check the directory area indicated by xxx. Overlapping directory entries is a possible cause of error.

**DIRECTORY ERROR CONTINUING READ;
STATUS =xxx AT DEVICE yyy.
CONTINUING SCAN**

Issued by: \$DISKUT1

Explanation: An error occurred while reading the directory of the specified device.

System Action: Continues processing.

User Response: Possibly, the door of the diskette drive is open, or the disk device is varied offline.

DIRECTORY I/O ERROR, DISK RC = xxx

Issued by: \$INITDSK

Explanation: The utility was unable to read or write to the directory of the disk. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. It may be necessary to reinitialize the volume or device in question.

DIRECTORY I/O ERROR RC = xxx

Issued by: \$COMPRES

Explanation: An I/O error occurred in reading the directory of a volume or device being compressed. The compress cannot continue.

System Action: Returns an error code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Then retry the function.

D

DIRECTORY OPEN ERROR. RETURN CODE

= xxx

Issued by: \$IAMUT1

Explanation: During the process of attempting to open the directory data set, an error occurred. The result was a \$DISKUT3 return code of xxx.

System Action: Terminates the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

DIRECTORY READ ERROR

Issued by: \$PREFIND

Explanation: An error occurred while reading the directory of the volume in use.

System Action: Returns to command mode.

User Response: Use a data set or program from another volume, or attempt to back up the damaged volume.

DIRECTORY READ ERROR. RETURN CODE

= xxx

Issued by: \$IAMUT1

Explanation: During the process of reading the directory data set, a read error occurred. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DIRECTORY WRITE ERROR. RETURN

CODE = xxx

Issued by: \$IAMUT1

Explanation: During the process of updating the directory data set (DE, IE, or UE subcommands of DR), a write error occurred or, during the process of adjusting the size of the directory data set (AS subcommand of DR), a write error occurred while writing to the new data set (\$WRKDIR). The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISCONNECTED FROM xxx

Issued by: \$MTM

Explanation: The terminal xxx is disconnected from the Multiple Terminal Manager.

System Action: Disconnects the terminal as requested by the user.

User Response: Reconnect the terminal if desired, or use it as you would any other terminal not connected to the Multiple Terminal Manager.

DISCONNECT FAILED FOR xxx,yyy

Issued by: \$IAMUT1

Explanation: A \$IAM DISCONNECT request was issued for the named indexed file and resulted in an error condition. The fields xxx,yyy stand for the data set name and volume in question.

System Action: Ends the operation. Displays a completion code.

User Response: Examine the \$IAMUT1 completion code and take the appropriate action.

DISK DIRECTORY NOT AVAILABLE

Issued by: \$INITDSK

Explanation: The utility was unable to access the disk directory. It is in use by another program or device.

System Action: Returns to command mode.

User Response: Retry several times. If necessary, wait until the device or volume is not in use.

DISK ERROR DURING CONCATENATION OPEN

Issued by: \$RJE2780, \$RJE3780

Explanation: While the system was trying to open a data set, an I/O error occurred.

System Action: Issues a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR DURING INITIALIZATION, RC=xxx

Issued by: \$MTM

Explanation: A disk error occurred while reading the SCRNS volume directory, the PRGRMS volume directory, the TERMINAL data set, or an error occurred while writing to the \$MTMSTORE data set. The result was a return code of xxx.

System Action: Aborts initialization.

User Response: Check the Multiple Terminal Manager return code to find the cause of the problem, and take the appropriate action.

DISK ERROR DURING OPEN - FUNCTION ABORTED

Issued by: \$RJE2780

Explanation: While attempting to open a data set, the system encountered an I/O error, and could not continue processing the requested function.

System Action: Issues a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR - PUNCHING ABORTED

Issued by: \$RJE2780, \$RJE3780

Explanation: While performing the punching function, the system encountered an I/O error. It could not continue punching.

System Action: Issues a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR RC = xxx - RECORD IGNORED, RESTORE PROCEEDING

Issued by: \$TAPEUT1

Explanation: An error occurred while writing a record to disk. At least one 256-byte area of the record will be ignored as the record is rewritten.

System Action: Continues processing.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR RC = xxx

Issued by: \$DICOMP

Explanation: During execution of the requested function, an error occurred in reading or writing to the disk. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR RC = xxx - RECORD NUMBER = yyy, SAVE PROCEEDING

Issued by: \$TAPEUT1

Explanation: During the save of a tape, an error occurred in writing data to a disk. The result was a return code of xxx.

System Action: Continues processing.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR READING SUBMIT FILE

Issued by: \$RJE2780, \$RJE3780

Explanation: While reading the submit file, the system encountered an I/O error.

System Action: Issues a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK ERROR WHILE SPOOLING

Issued by: \$RJE2780, \$RJE3780

Explanation: While performing the spooling function, the system encountered an I/O error.

System Action: Issues a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISKETTE DATA SET TOO SMALL

Issued by: \$COPY

Explanation: The specified data set is too small to hold a copy of the source data.

System Action: Awaits operator response.

User Response: Use another, larger diskette data set, or a disk data set, to hold the copy.

DISKETTE I/O ERROR

Issued by: \$IMAGE

Explanation: An error occurred during a read or write of a diskette.

System Action: Supplies a return code. Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISKETTE IS ONE SIDED - CANNOT BE DOUBLE DENSITY

Issued by: \$DASDI

Explanation: You attempted to define a single-density, one-sided diskette as double-density.

System Action: Ends the operation.

User Response: Define the one-sided diskette as single-density, or use a two-sided double-density diskette (type 2D) and define it as double density.

DISKETTE: xxx IS THE WRONG TYPE

Issued by: \$MOVEVOL

Explanation: The diskette you specified is invalid for use with the \$MOVEVOL utility.

System Action: Prompts for another diskette.

User Response: Specify a diskette of the proper type and retry the function.

DISKETTE NOT SUPPORTED FOR xxx FUNCTION

Issued by: \$TAPEUT1

Explanation: The use of a diskette is unacceptable for either the save or restore function. The xxx identifies the function in question.

System Action: Ends the operation.

User Response: Specify a disk device for use by the function.

DISK(ETTE) READ ERROR RC = xxx

Issued by: \$TERMUT2

Explanation: An error occurred while reading from the input data set. The result was a return code of xxx.

System Action: Returns to command mode.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Recover the input data set and retry the function.

D

DISKETTE READ ERROR, RETRY (Y/N)?

Issued by: \$MOVEVOL

Explanation: A read error occurred on a diskette, causing the function to fail.

System Action: Ends the operation.

User Response: Retry. If problem persists, use another diskette.

DISK(ETTE) WRITE ERROR RC = xxx

Issued by: \$TERMUT2

Explanation: An error occurred while writing to the output data set. The result was a return code of xxx.

System Action: Returns to command mode.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Save to another data set.

DISK HAS NOT HAD AN ID DONE YET

Issued by: \$INITDSK

Explanation: You tried to access a device that has not been initialized for EDX.

System Action: Returns to command mode.

User Response: Initialize the device with \$INITDSK, or retry the function using another device.

DISK INITIALIZATION TERMINATED ABNORMALLY - RC = xxx

Issued by: \$DASDI

Explanation: An error occurred during disk initialization, resulting in the termination of the utility. The return code was xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISK I/O ERROR xxx

Issued by: \$COPYUT1

Explanation: During the copy function that you requested, an error occurred in either reading or writing to the disk. The result was a return code of xxx.

System Action: Ends the operation. For a multiple member copy, continues copying other members.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**DISK I/O ERROR ATTEMPTING TO READ
\$IAM PROGRAM. RETURN CODE = xxx**

Issued by: \$IAMUT1

Explanation: The program was attempting to read the \$IAM program to obtain current storage information. The read failed. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**DISK I/O ERROR ATTEMPTING TO WRITE
\$IAM PROGRAM. RETURN CODE = xxx**

Issued by: \$IAMUT1

Explanation: The program was attempting to write the new storage information back out to the \$IAM program. The write failed. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**DISK I/O ERROR IS xxx FOR DATA SET
yyy,zzz AT RECORD nnn**

Issued by: \$HXUT1

Explanation: The utility detected an I/O error during a copy operation (RX or WX commands). The result was a return code of xxx. The data set name and volume are yyy,zzz. The record where the error occurred is nnn.

System Action: Displays information on the next exchange record and the next EDX record. The utility then asks whether you want to continue or stop the copy operation and waits for your response.

User Response: Check the disk and diskette read/write return code to determine the problem and whether to continue the operation. Enter "Y" to stop the copy operation. Enter "N" to continue; the content of the copied record is undefined. To examine the copied record, use the dump facility of \$DISKUT2.

**DISK I/O ERROR RC = xxx IN DATA SET
yyy,zzz**

Issued by: \$DISKUT2, \$MSGUT1

Explanation: A disk read or write error was discovered in the named data set.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

D

DISK I/O ERROR RC = xxx IN THE JOB QUEUE DATA SET

Issued by: \$SUBMIT, \$JOBQUT

Explanation: The utility could not get the proper information from the job queue data set.

System Action: Returns to command mode.

User Response: Initialize the job queue data set with the IJQ command and retry the function.

Note: this process will delete all jobs that were on the queue.

DISK I/O ERROR READING NUCLEUS, STATUS = xxx

Issued by: \$\$S1S1UT1

Explanation: An error occurred while reading the disk. The result was a return code of xxx.

System Action: Returns to command mode.

User Response: Recreate the nucleus or retry the function using a different nucleus.

DISK I/O ERROR READING "IPLPGM", STATUS = xxx

Issued by: \$\$S1S1UT1

Explanation: "IPLPGM" is the bootstrap sent to the remote processor for an IPL function. The actual name of the program is IPLS1S1.

System Action: Returns to command mode.

User Response: Verify that IPLS1S1 is on disk and retry the IPL function.

DISK I/O ERROR WRITING DISK DIRECTORY POINTER. RC =xxx

Issued by: \$INITDSK

Explanation: The new device directory location has a read/write error.

System Action: Returns to command mode.

User Response: Retry the function and specify a different location, or the default.

DISK IS IN USE BY ANOTHER PROGRAM

Issued by: \$COMPRES,\$DISKUT1,\$COPYUT1

Explanation: While trying to perform the function, the system found the target disk busy.

System Action: Terminates the function or returns to command mode.

User Response: Be sure that no one else is using the volume or device and retry the function or device.

DISK READ ERROR

Issued by: \$MTM

Explanation: An internal Multiple Terminal Manager disk read error has occurred.

System Action: Results are unpredictable.

User Response: Determine the cause of the error and retry operation.

DISK READ ERROR (DIRECTORY)

Issued by: \$UPDATE, \$UPDATEH

Explanation: An error occurred while reading the directory of the volume in use.

System Action: Returns to command mode.

User Response: Restore the volume and retry the function.

DISK READ ERROR (OUTPUT PROGRAM)

Issued by: \$UPDATE, \$UPDATEH

Explanation: A read error occurred on the output program data set.

System Action: Returns to command mode.

User Response: Recreate output program data set and retry the function.

DISK READ ERROR RC = xxx

Issued by: \$COPY

Explanation: During the copy function, an error occurred while reading from the disk. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

\$DISKUT3 ERROR IS xxx

Issued by: \$HXUT1

Explanation: The \$DISKUT3 operation (on the target data set) requested by \$HXUT1 failed.

System Action: The command terminates.

User Response: Check the \$DISKUT3 return code to find the cause of the problem and take the appropriate action.

\$DISKUT3 LOAD ERROR ENCOUNTERED

Issued by: \$VERIFY

Explanation: \$VERIFY was unable to load \$DISKUT3.

System Action: An ABEND message follows with the return code from LOAD contained in it. \$VERIFY terminates.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action. If the LOAD return code indicates insufficient storage, rerun \$VERIFY when more space is available in the current partition.

D

\$DISKUT3 LOAD ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: During the process of loading \$DISKUT3 to perform an open, delete, rename or allocate operation, an error occurred.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

DISK WRITE ERROR (DIRECTORY)

Issued by: \$UPDATE, \$UPDATEH

Explanation: An error occurred while writing to the directory in use.

System Action: Returns to command mode.

User Response: Check the directory to see that the member you were writing to exists. If the directory cannot be accessed in any way, you will have to restore the volume.

DISK WRITE ERROR (OUTPUT PROGRAM)

Issued by: \$UPDATE, \$UPDATEH

Explanation: A write error occurred on the output program data set.

System Action: Returns to command mode.

User Response: Reallocate the output program data set and retry the function.

DISK WRITE ERROR RC = xxx

Issued by: \$COPY

Explanation: During the copy function, an error occurred while writing to the disk. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DISPLAY DATA BASE - DATA INVALID

Issued by: \$DIINTR

Explanation: The data set in use contains invalid items. The data set is not a valid partitioned data set.

System Action: Ends the operation.

User Response: Rebuild the input data set.

DO NOT RECOGNIZE THIS PROCESSOR TYPE: xxx

Issued by: \$STGUT1

Explanation: The utility cannot perform the requested function on the model of processor in your system.

System Action: Returns to command mode.

User Response: Re-IPL to try to correct the problem. If the same message keeps occurring, you cannot use \$STGUT1 with your processor.

DO NOT SPECIFY BOTH MODNAME AND DSX

Issued by: \$EDXASM

Explanation: Both MODNAME and DSX were specified on the COMP instruction, and only one is allowed.

System Action: The error is flagged and compilation continues.

User Response: Specify either MODNAME or DSX on the COMP instruction, but not both.

DO WHILE OR DO UNTIL MUST HAVE EVEN NUMBER OF OPERANDS

Issued by: \$EDXASM

Explanation: An odd number of operands were specified.

System Action: The error is flagged and compilation continues.

User Response: Ensure that an even number of operands is compared to each other.

DS COMMAND NOT PRECEDED BY PROGRAM COMMAND

Issued by: \$JOBUTIL

Explanation: In the procedure data set, you have entered information in the wrong order.

System Action: Ends the utility.

User Response: Edit the procedure data set and place all the information in the proper order. Then retry the utility.

DS#= OPERAND MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The name used to refer to a data set control block was omitted on the DSCB instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify a 1 - 8 character name for the DS#= operand.

DSLEN= OPERAND IS INVALID

Issued by: \$EDXASM

Explanation: The size for the direct access space was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Define the value for DSLEN= explicitly.

DSNAME= OPERAND MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The data set name for the data set control block to be accessed was omitted on the DSCB instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the 1-8 character name of the data set for the DSNAME= operand.

D

DSOPEN ERROR - DATA SET NOT FOUND

Issued by: \$TAPEUT1

Explanation: You specified an invalid data set name.

System Action: Returns to command mode.

User Response: Specify a valid data set name.

DSOPEN ERROR - INVALID IODA

Issued by: \$TAPEUT1

Explanation: The system tried to open a data set on a device that doesn't exist. The device name you specified is invalid.

System Action: Ends the operation.

User Response: Specify a data set on a valid device.

DSOPEN ERROR - I/O ERROR RC = xxx

Issued by: \$TAPEUT1

Explanation: A disk read error prevented the data set from being opened.

System Action: Returns to command mode.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DSOPEN ERROR - VOLUME NOT FOUND

Issued by: \$TAPEUT1

Explanation: You specified an invalid volume name.

System Action: Returns to command mode.

User Response: Specify a valid volume name and retry.

DUMP CANCELLED. DISK I/O ERROR RC = xxx

Issued by: \$BSCUT1

Explanation: An error occurred while the trace data was being dumped to the trace file. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

DUMP RANGE INVALID VALID RANGE IS xxx TO yyy

Issued by: \$DUMP

Explanation: An invalid starting or ending storage address was entered during a partial storage dump.

System Action: Terminates the \$DUMP utility.

User Response: Enter a valid storage address in the range xxx to yyy and restart the \$DUMP utility.

**DUPLICATE NAME IN DIRECTORY - ADD
ABORTED**

Issued by: \$DICOMP

Explanation: The new name you specified with the ADD command is the same as an existing name.

System Action: Returns to command mode.

User Response: List all existing names with \$DIUTIL. Retry the function with a new, unique name.

**DUPLICATE VOLUME LABELS ARE NOT
ALLOWED**

Issued by: \$COPY

Explanation: You tried to copy to and from the same volume.

System Action: Returns to command mode.

User Response: Specify different source and target volumes and retry.

E

ECB NAME IS NUMERIC - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The ECB operand on the CAxxx instruction is numeric. The first character must be alphabetic.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a valid ECB operand whose first character is alphabetic, and reassemble.

ECB NAME TOO LONG - CAxxx TERMINATED

Issued by: \$S1ASM

Explanation: The ECB operand on the CAxxx instruction has more than eight characters.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify an ECB operand that is less than or equal to eight characters in length, and reassemble.

ECB NAME IS NUMERIC

Issued by: \$S1ASM

Explanation: The ECB operand on the previous CAxxx instruction is numeric. The first character must be alphabetic.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid ECB operand whose first character is alphabetic, and reassemble.

ECB NOT SPECIFIED - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The ECB operand on the CAxxx instruction is a required operand but is not specified.

System Action: Terminates the expansion.

User Response: On the CAxxx instruction, specify a valid ECB operand, and reassemble.

ECB NAME NOT SPECIFIED - CAxxx TERMINATED

Issued by: \$S1ASM

Explanation: The ECB operand on the CAxxx instruction is a required operand but is not specified.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid ECB operand, and reassemble.

EITHER YES= OR NO= MUST BE CODED**Issued by:** \$EDXASM**Explanation:** A “YES=” or “NO=” operand must be coded on the QUESTION instruction, or the parameter for a keyword operand must be “YES” or “NO”.**System Action:** The error is flagged and compilation continues.**User Response:** Specify either “YES=” or “NO=” operand on the QUESTION instruction or as the value of a keyword operand requiring such a response.**ENABLE SPECIFIED INCORRECTLY - YES ASSUMED****Issued by:** \$EDXASM**Explanation:** A value other than “YES” or “NO” was entered for the ENABLE= operand on the CATRACE instruction.**System Action:** A value of “YES” is the default for the ENABLE= operand on the CATRACE instruction.**User Response:** If “NO” is required for the ENABLE= operand on the CATRACE instruction, specify such, and reassemble.**END OF FILE ERROR (OBJECT MODULE)****Issued by:** \$UPDATE**Explanation:** \$UPDATE found an end of file (EOF) but found no end record. An error may have occurred during the compilation of the object module.**System Action:** Terminates the \$UPDATE utility.**User Response:** Recompile the object module and retry the utility.**END OF FILE ERROR (OUTPUT PROGRAM)****Issued by:** \$UPDATE, \$UPDATEH**Explanation:** An attempt to write to the output program data set failed because the end of file was encountered. The output file was too small.**System Action:** Returns to command mode.**User Response:** Allocate a larger program data set and retry the function.**END OF INPUT D/S****Issued by:** \$IAMUT1**Explanation:** Using the LO, UN, or RO commands, an end of data condition was encountered on the input data set.**System Action:** If the command is LO, a prompt is issued for another input data set. If the command is UN or RO, the function ends normally.**User Response:** Respond to the prompt for another input data set if desired.

**END OF VOLUME ENCOUNTERED BEFORE
VERIFY COMPLETE**

Issued by: \$COMPRES

Explanation: Either you allocated too much dynamic storage when you loaded the utility, or the directory to be compressed is unusable due to an I/O error.

System Action: Ends the operation.

User Response: In the first case, load the utility with less (or none) dynamic storage. In the second case, restore the volume being compressed.

**ENDRJE IN PROGRESS - SUBMIT/X NOT
ALLOWED**

Issued by: \$RJESNA

Explanation: You tried to submit a job stream after the ENDRJE command was entered.

System Action: Continues \$RJESNA termination.

User Response: Wait for \$RJESNA to terminate, then restart \$RJESNA and submit the job.

**ENTER DSNAME,VOLUME, OR ALL FOR
ALL SEC. FILES**

Issued by: \$VERIFY

Explanation: During the secondary index verification, the program requests the name of the secondary index file to be cross verified against the primary file just verified.

System Action: Awaits operator response.

User Response: If you want to cross verify the secondary index file, enter the data set name and volume name of the secondary file. Enter "ALL" to cross verify all secondary files.

**EOD ERROR DETECTED
ACTUAL DEV/VOL NOT EQUAL TO
INDICATED LENGTH
EXPECTED = xxx ACTUAL = yyy**

Issued by: \$TAPEUT1

Explanation: An EOD was detected unexpectedly while reading the disk.

System Action: Returns to command mode.

User Response: Rerun the function using the actual length specified.

EOD ON DISK - DEVICE NOT RESTORED**Issued by:** \$TAPEUT1**Explanation:** You attempted to restore a disk device, volume, or data set from tape. The disk area you are restoring, however, is not large enough to hold the data on the tape. The system encountered an end-of-data condition while it was writing to the disk area.**System Action:** The system ends the operation.**User Response:** Ensure that the device, volume, or data set being restored is the same size as or larger than the device, volume, or data set saved on tape.**ERROR - A COMMON SECTION AND AN OVERLAY CONTROL SECTION MATCHED - xxx****Issued by:** \$EDXLINK**Explanation:** A CSECT xxx specified in one of the overlay segments matched a COMMON section.**System Action:** Doesn't include the indicated CSECT in the generated program. \$EDXLINK sets a completion code of 8.**User Response:** Resolve the conflict and relink. Perhaps the object module containing the CSECT was included in an overlay segment when it should have belonged in the root segment.**ERROR xxx AT RECORD yyy****Issued by:** \$INITDSK**Explanation:** The verify function has detected an error.**System Action:** Processing continues.**User Response:** When processing is finished, isolate the defective area and correct the problem, if possible.**ERROR ALLOCATING NUCLEUS, \$DISKUT3 RC = xxx****Issued by:** \$XPSLINK from \$XPSPOST**Explanation:** An error occurred while trying to allocate the nucleus on disk or diskette.**System Action:** Terminates function or prompts for a retry.**User Response:** Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.**ERROR ATTEMPTING TO OPEN \$IAM. RETURN CODE = xxx****Issued by:** \$IAMUT1**Explanation:** The program attempted to open the \$IAM program to obtain current storage information. The open failed, resulting in a return code of xxx.**System Action:** Ends the operation.**User Response:** Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

**ERROR - BLANK LINE NOT ALLOWED
STATEMENT IGNORED**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The current line in the link control data set is blank.

System Action: Ends the operation.

User Response: Remove the blank line from the link control data set.

**ERROR - CANNOT RESTORE A DEVICE
FROM A xxx SAVE TAPE**

Issued by: \$TAPEUT1

Explanation: Using the RT command, a device restore was requested, but the tape only contained a volume or data set.

System Action: Terminates the utility.

User Response: Request a volume or data set restore to retrieve the data on the tape.

**ERROR - CANNOT RESTORE A xxx FROM
A yyy SAVE TAPE**

Issued by: \$TAPEUT1

Explanation: Using the RT command, a device restore was requested, but the type of disk being restored (xxx) does not match the type of disk that was saved on tape (yyy).

System Action: Terminates the utility.

User Response: To retrieve the data on the tape, restore the volumes individually.

**ERROR - CANNOT RESTORE A VOLUME
FROM A DATASET SAVE TAPE**

Issued by: \$TAPEUT1

Explanation: Using the RT command, a volume restore was requested; however, the tape only contained a data set.

System Action: Terminates the utility.

User Response: Request a data set restore to retrieve the data on the tape.

**ERROR - DATASET NAME CANNOT BE
CURRENT NUCLEUS NAME**

Issued by: \$XPSLINK from \$XPSPOST

Explanation: You tried to create a nucleus with the same name and volume as the nucleus currently IPLed. This is not allowed.

System Action: Terminates function or prompts for retry.

User Response: Retry with a different nucleus name or volume.

ERROR - DATASET NAME TOO LONG - xxx

Issued by: \$EDXLINK

Explanation: The length of the data set name is more than eight characters.

System Action: Data set is not opened; \$EDXLINK continues and sets a completion code of 8.

User Response: Re-enter the data set name with a maximum of 8 characters.

**ERROR DELETING EXISTING NUCLEUS,
\$DISKUT3 RC = xxx.**

Issued by: \$XPSLINK from \$XPSPOST

System Action: Terminates function or prompts for retry.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

**ERROR - DUPLICATELY DEFINED SYMBOL
- xxx IN yyy,zzz**

Issued by: \$EDXLINK

Explanation: Two externally defined symbols have the same name, xxx. They are located in data set yyy on volume zzz.

System Action: \$EDXLINK discards the second symbol and any references to it are redirected to the first symbol. Also sets a completion code of 8.

User Response: You may have included the same object module twice. If so, edit the control statement data set and remove the INCLUDE statement that refers to the extra object module. You may have given two different externally defined symbols the same name. If so, rename one of these externally defined symbols and relink.

ERROR - EDXINIT WAS NOT INCLUDED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: EDXINIT was not included in the link control data set and is required.

System Action: Ends the operation.

User Response: Include EDXINIT in the link control data set.

ERROR - EDXSTART WAS NOT INCLUDED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: EDXSTART was not included in the link control data set and is required.

System Action: Ends the operation.

User Response: Include EDXSTART in the link control data set.

ERROR - EDXSYS WAS NOT INCLUDED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: EDXSYS was not included in the link control data set and is required.

System Action: Ends the operation.

User Response: Include EDXSYS in the link control data set.

E

ERROR ENCOUNTERED DURING CLOSE OF INDEXED ACCESS METHOD

xxx,yyy, ERROR CODE=zzz

Issued by: \$MTM

Explanation: An error occurred during AUTOCLOSE of an Indexed Access Method data set named by xxx,yyy. The result was a return code of zzz.

System Action: None.

User Response: Check the Indexed Access Method return code to find the cause of the problem and take the appropriate action.

ERROR ENCOUNTERED USING \$DISKUT3 FOR OPEN

Issued by: \$VERIFY

Explanation: \$VERIFY attempted to open the indexed file and a \$DISKUT3 error occurred.

System Action: An abend message follows with the return code from \$DISKUT3 contained in it. \$VERIFY terminates.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

ERROR: END OF XPSTABLE NOT ENCOUNTERED

Issued by: \$XPSLINK from \$XPSPOST

Explanation: The XPSTABLE is not in the correct position. This is a fatal error to \$XPSPOST.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Run \$XPSLINK again.

ERROR - EOF REACHED IN WORK DATA SET - xxx,yyy

Issued by: \$EDXLINK

Explanation: The work data set, xxx,yyy, which you supplied for \$EDXLINK, was not large enough to complete the required link-edit.

System Action: Terminates processing of the link edit and ends \$EDXLINK. Also sets a completion code of 16.

User Response: Restart \$EDXLINK specifying a larger work data set.

**ERROR EXIT INCORRECTLY SPECIFIED -
NONE ASSUMED**

Issued by: \$S1ASM

Explanation: The ERROR= operand on the CAxxx instruction is incorrectly specified.

System Action: Does not assume any error exit in the expansions.

User Response: Either eliminate the ERROR= operand on the CAxxx instruction, or specify the operand correctly, and reassemble.

**ERROR EXPECTED RC = xxx ACTUAL RC
= yyy**

Issued by: \$TAPEUT1

Explanation: While exercising the tape, the utility expected to receive a certain return code, and actually received another code.

System Action: Ends the operation.

User Response: Check the actual tape read/write return code to find the cause of the problem, and take appropriate action.

**ERROR - INCOMPLETE CONTROL
STATEMENT DATA SET**

Issued by: \$EDXLINK

Explanation: No LINK statement was found in the primary control statement data set.

System Action: Terminates because \$EDXLINK didn't know where to put the program you were trying to generate. Also sets a completion code of 16.

User Response: Edit the primary control statement data set. You must include a LINK statement to identify where to put the program. Then relink.

**ERROR - INSUFFICIENT CONTIGUOUS
SPACE ON THE VOLUME xxx**

Issued by: \$EDXLINK

Explanation: There is not enough contiguous space on the volume to create the output program data set.

System Action: If \$EDXLINK is invoked interactively or via "\$L", a prompt will be issued for different volume name. Otherwise, \$EDXLINK will print the map, but no program will be stored. \$EDXLINK also sets a completion code of 12.

User Response: Compress the volume using the \$COMPRES utility.

E

ERROR - INSUFFICIENT CONTIGUOUS SPACE ON THE VOLUME xxx COMPRESS VOLUME * REQUIRED CONTIGUOUS SPACE MUST BE xxx RECORDS**

Issued by: \$XPSLINK from \$XPSPOST

Explanation: There is not enough contiguous space on the volume in use to perform the requested function.

System Action: Terminates function or prompts for retry.

User Response: Compress the volume using the \$COMPRES utility; then retry.

ERROR - INSUFFICIENT DIRECTORY SPACE ON THE VOLUME xxx

Issued by: \$EDXLINK

Explanation: There is not enough directory space on the volume to create the output program data set.

System Action: If \$EDXLINK is invoked interactively or via "\$L", a prompt will be issued for different volume name. Otherwise, \$EDXLINK will print the map, but no program will be stored. \$EDXLINK also sets a completion code of 12.

User Response: Specify a volume with a greater amount of directory space.

ERROR - INSUFFICIENT DIRECTORY SPACE ON THE VOLUME xxx

Issued by: \$XPSLINK from \$XPSPOST

Explanation: There was not a directory entry available for the nucleus.

System Action: Terminates function or prompts for retry.

User Response: Create a free directory entry on the volume originally specified by deleting a member, or specify a different volume, and try again.

ERROR - INSUFFICIENT DYNAMIC STORAGE WHILE PROCESSING xxx,yyy

Issued by: \$EDXLINK

Explanation: There was not enough dynamic storage to complete the link.

System Action: Terminates \$EDXLINK after message is printed, and sets a completion code of 16.

User Response: Use the Set Storage (SS) option of \$DISKUT2 to set storage to a greater value for \$EDXLINK.

ERROR - INSUFFICIENT SPACE ON THE VOLUME xxx

Issued by: \$EDXLINK

Explanation: There is not sufficient space on the indicated volume for the output of \$EDXLINK.

System Action: If \$EDXLINK is invoked interactively or via "\$L", a prompt will be issued for different volume name. Otherwise, \$EDXLINK will print the map, but no program will be stored. \$EDXLINK also sets a completion code of 12.

User Response: Specify a volume with a greater amount of space.

ERROR - INSUFFICIENT SPACE ON THE VOLUME xxx *REQUIRED CONTIGUOUS SPACE MUST BE xxx RECORDS**

Issued by: \$XPSLINK from \$XPSPOST

Explanation: The volume in use does not have enough free space to contain the nucleus.

System Action: Terminates function or prompts for retry.

User Response: Create more free space on the volume originally specified by deleting members, or use a different volume, and try again.

ERROR - INVALID DATASET NAME - xxx

Issued by: \$EDXLINK

Explanation: Data set xxx is one of the following invalid names: \$\$EDXLIB, \$\$EDXVOL, \$\$, or \$DDExx, (where xx is the device address).

System Action: Data set is not opened; \$EDXLINK continues and sets a completion code of 8.

User Response: Change the name of the data set to a valid name.

ERROR - INVALID ESD ITEM IN xxx.yyy

Issued by: \$EDXLINK

Explanation: \$EDXLINK encountered an ESD type it does not support. \$EDXLINK only supports the following types: Section Definitions (CSECT), Label Definitions (ENTRY), External References (EXTRN), Private Code (unlabeled CSECT), Common Sections (COMMON), Weak External References, (WXTRN), and Global (GLOBAL).

System Action: The indicated object module will not be included in the generated program. Also sets a completion code of 8.

User Response: Look at the indicated object module to see whether it is a valid object module. If it is not, remove or replace it in your control statement data set. If it is, you are not permitted to link this object module with \$EDXLINK.

E

ERROR - INVALID OBJECT MODULE RECORD IN xxx,yyy

Issued by: \$EDXLINK

Explanation: The indicated object module contained an invalid object module record.

System Action: Dumps 16 bytes starting at the location where the error occurred in the object module. The system will not include this object module in the program generated. Also sets a completion code of 8.

User Response: You must regenerate the indicated object module. To do this, assemble or compile the source that generated the indicated object module and relink.

ERROR - INVALID OBJECT RECORD IN xxx,yyy

Issued by: \$EDXLINK

Explanation: This data set contains an invalid object record.

System Action: Halts processing of the object module; \$EDXLINK continues with remaining modules, and sets a completion code of 8.

User Response: Reassemble or recompile the program and verify that it completes successfully.

ERROR INVALID PARTITION NUMBER xxx

Issued by: \$SUBMIT, \$JOBQUT

Explanation: The partition number you entered is not in use in your system.

System Action: Prompts for another partition number.

User Response: Enter a valid partition number. To determine the partitions in your system, issue the \$A ALL command.

ERROR - INVALID PARTITION NUMBER. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The PART statement contains an invalid partition number (either a number outside the range 1 - 8, or a partition number that does not exist in your system).

System Action: Ends the operation.

User Response: Enter a valid partition number.

ERROR - INVALID STATEMENT. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The current line in the link control data set contains an invalid statement (something other than IN, LI, PA, CO, AU, or EN). The line is printed for you to examine.

System Action: Ends the operation.

User Response: Remove the invalid entry from the line.

ERROR - INVALID SYNTAX - NO DATASET NAME SPECIFIED

Issued by: \$EDXLINK

Explanation: No data set name was entered on the INCLUDE control statement.

System Action: Statement ignored; \$EDXLINK continues, and sets a completion code of 8.

User Response: Enter a data set before the volume name.

ERROR - INVALID SYNTAX. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The current line in the link control data set contains invalid syntax. The line is printed for you to examine.

System Action: Ends the operation.

User Response: Correct the invalid syntax in the line.

ERROR - I/O ERROR IN WORK DATA SET - xxx,yyy RC=zzz

Issued by: \$EDXLINK

Explanation: \$EDXLINK received an I/O error other than an EOF (RC=10) while reading or writing the work data set.

System Action: Terminates processing of the link edit and ends \$EDXLINK. Also sets a completion code of 16.

User Response: Restart \$EDXLINK specifying a different work data set.

ERROR - I/O ERROR READING DIRECTORY OF VOLUME xxx

Issued by: \$EDXLINK

Explanation: A hardware error occurred while reading the volume directory.

System Action: Data sets on the volume will not be opened; \$EDXLINK continues, but the program will not be stored. \$EDXLINK also sets a completion code of 12.

User Response: Investigate and correct the hardware error.

ERROR - I/O ERROR WRITING DIRECTORY OF xxx

Issued by: \$EDXLINK

Explanation: A write error occurred while writing the volume directory.

System Action: The data set will never be created; no output will be stored from \$EDXLINK.

User Response: Allocate the data set in a different place.

E

ERROR LOADING \$DISKUT3 - LOAD RETURN CODE IS xxx

Issued by: \$HXUT1

Explanation: \$HXUT1 attempted to load \$DISKUT3 and the load operation failed. The result was a return code of xxx.

System Action: The command terminates.

User Response: Check the LOAD return code to find the cause of the problem and take the appropriate action.

ERROR - MODULE NAME TOO LONG. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The module name specified is greater than eight characters in length.

System Action: Ends the operation.

User Response: Specify a module name of the correct length.

ERROR - MUST BE IN PARTITION ONE. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The module name specified was included in a partition other than partition one; that module is allowed only in partition one.

System Action: Ends the operation.

User Response: Move the module to partition one.

ERROR - NO OBJECT MODULE END RECORD IN xxx,yyy

Issued by: \$EDXLINK

Explanation: The indicated object module is not complete because no END record is specified.

System Action: Will not include this object module in the program generated. Sets a condition code of 8.

User Response: Re-create the indicated object module. To do this, assemble or compile the source that generated the indicated object module and relink.

ERROR - NO PROGRAM HEADER IN FIRST MODULE

Issued by: \$EDXLINK

Explanation: The first data set included in this link does not contain a program header.

System Action: \$EDXLINK does not put start address of program into program header (needed by loader) and sets a completion code of 8.

User Response: Relink, specifying an object module with a program header as the first INCLUDE.

ERROR - xxx,yyy NOT DATA TYPE

Issued by: \$EDXLINK

Explanation: The data set requested is not a data type data set.

System Action: Data set is not opened; \$EDXLINK continues and sets a completion code of 8.

User Response: Re-create the data set to be data type.

ERROR - xxx NOT INITIALIZED

Issued by: \$EDXLINK

Explanation: The volume requested is not initialized.

System Action: Volume is not opened; \$EDXLINK continues, and sets a completion code of 8.

User Response: Initialize the volume using the \$INITDSK utility.

ERROR - xxx NOT FOUND ON yyy

Issued by: \$EDXLINK

Explanation: The data set requested does not exist on the volume specified.

System Action: Data set is not opened; \$EDXLINK continues, and sets a completion code of 8.

User Response: Allocate the data set on the volume or correct the volume name.

ERROR - NO TM DETECTED

Issued by: \$TAPEUT1

Explanation: While reading from the tape, the utility did not find a tape mark it expected.

System Action: Continues processing.

User Response: Wait for processing to complete.

ERROR - xxx NOT FOUND IN DIRECTORY ON TAPE

Issued by: \$TAPEUT1

Explanation: Using the RT command, a volume or data set restore was requested, but the volume or data set xxx was not found on the tape.

System Action: Terminates the utility.

User Response: Run the utility again. Specify the correct name of the volume or data set.

ERROR - xxx NOT MOUNTED

Issued by: \$EDXLINK

Explanation: The volume requested is not mounted or does not exist.

System Action: Volume is not opened; \$EDXLINK continues and sets a completion code of 8.

User Response: Mount the volume or specify the new volume name.

E

ERROR OCCURRED DURING DISK READ

Issued by: \$VERIFY

Explanation: \$VERIFY attempted to read from the indexed file and an unrecoverable disk read error occurred.

System Action: An abend message follows with the return code from READ contained in it. \$VERIFY terminates.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

ERROR ON TARGET - RC = xxx

Issued by: \$TAPEUT1

Explanation: The device you are copying to detected an error condition.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

ERROR - xxx ON TAPE IS LARGER THAN ALLOCATED DISK SPACE MINIMUM SIZE REQUIRED IS yyy RECORDS

Issued by: \$TAPEUT1

Explanation: Using the RT command, a volume or data set (xxx) restore was requested, but the target volume or data set is too small.

System Action: Terminates the utility.

User Response: Reallocate the volume or data set with at least yyy records and run the utility again.

ERROR - PARTITIONS CAN NOT BE SPLIT

Issued by: \$XPSLINK from \$XPSPRE

Explanation: A partition containing supervisor support is split in the link control data set.

System Action: Ends the operation.

User Response: Group together all support for the same partition. Partition one must be first.

ERROR - READ ERROR ENCOUNTERED ON xxx,yyy

Issued by: \$EDXLINK

Explanation: A read error occurred while processing this data set.

System Action: Terminates processing of the data set; \$EDXLINK continues with remaining data sets, and sets a completion code of 8.

User Response: Allocate the data set in a different space.

ERROR READING LOG CONTROL RECORD

Issued by: \$LOG

Explanation: A read error occurred on the log data set.

System Action: Returns to command mode.

User Response: End \$LOG and use a different log data set or use \$DISKUT1 to allocate a new one. Then reload \$LOG with the new log data set.

ERROR - REPLACE NOT SPECIFIED AND PROGRAM EXISTS

Issued by: \$EDXLINK

Explanation: There was no key word REPLACE on the LINK statement and the program exists.

System Action: If in interactive mode or loaded through "\$L", \$EDXLINK will prompt for replace or new output data set name. Otherwise, \$EDXLINK will print the map, but no program will be stored. \$EDXLINK sets a completion code of 12.

User Response: Change the name of the program on the LINK statement or add the operand "REPLACE" to the LINK statement.

ERROR RETURN CODE = xxx

Issued by: \$VERIFY

Explanation: \$VERIFY encountered an error such that it was unable to continue processing. The result was a return code of xxx. A previous message gives the reason for the termination.

System Action: Ends the operation.

User Response: Check the Indexed Access Method return code and take the appropriate action. Then retry the operation.

ERROR: SUPERVISOR NOT DEFINED FOR PARTITION 1

Issued by: \$XPSLINK from \$XPSPOST

Explanation: While running \$XPSPOST by itself (not under \$XPSLINK), you specified a work volume that does not contain XPS segments. It could also be that XPS segment 1 was deleted before it was used.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Use the work volume originally used by \$XPSLINK and \$XPSPRE, and run \$XPSLINK again.

ERROR - TAPE NOT COMPATIBLE WITH RESTORE UTILITY

Issued by: \$TAPEUT1

Explanation: Using the RT command, the SAVE function (\$TAPEUT1) used to create the tape is not the same version as the one being used to restore. For example, both the SAVE and RESTORE functions must be run under EDX V4.

System Action: Terminates the utility.

User Response: If the utility version used to create the tape is not known, you can find it by dumping the first record of the tape and examining the seventh word in the 18-byte control block. This word contains the version number of the tape.

ERROR - THE FOLLOWING MODULE WAS INCLUDED MORE THAN ONCE. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The module specified was already included once in the link control data set.

System Action: Ends the operation.

User Response: Delete the second statement that includes the specified module.

ERROR - THE WORK DATA SET xxx,yyy MUST BE AT LEAST 400 RECORDS

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The work data set specified contains less than 400 records.

System Action: Ends the operation.

User Response: Reallocate the work data set to contain at least 400 records, or use a different work data set with 400 or more records.

ERROR - UNABLE TO ALLOCATE xxx,yyy RC = zzz

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The specified data set was not allocated with \$DISKUT3.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

ERROR - UNABLE TO LOAD \$DISKUT3 FROM LOAD OR IPL VOLUME RC =xxx

Issued by: \$XPSLINK from \$XPSPRE and \$XPSPOST

Explanation: \$DISKUT3 was not found on the load or IPL volumes, or some other LOAD error occurred.

System Action: Ends the operation.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action.

**ERROR - UNABLE TO OPEN xxx.yyy
RC = xxx**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The specified data set could not be opened.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

**ERROR - UNABLE TO SET END OF DATA
FOR xxx.yyy RC = zzz**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: EOD was not set with \$DISKUT3 for the specified data set.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action.

**ERROR - UNRESOLVED EXTERNAL
REFERENCES**

Issued by: \$EDXLINK

Explanation: One or more external references remain unresolved after completion of the link edit.

System Action: Lists all the unresolved external references that remained after completion of the link. \$EDXLINK sets a completion code of 4.

User Response: Re-edit the control statement data set and include the necessary statements (INCLUDE or AUTOCALL) to resolve external references. Then relink.

ERROR - VOLUME xxx DOES NOT EXIST

Issued by: \$XPSLINK from \$XPSPPOST

Explanation: In response to a prompt for a new nucleus name and volume, the volume you specified could not be found.

System Action: Terminates function or prompts for retry.

User Response: Specify an existing volume and try again.

ERROR - VOLUME NAME TOO LONG - xxx

Issued by: \$EDXLINK

Explanation: Length of volume name is more than six characters.

System Action: Volume and data sets are not opened; \$EDXLINK continues.

User Response: Re-enter volume name with a maximum of 6 characters.

E

ERROR - VOLUME NAME TOO LONG. STATEMENT IGNORED

Issued by: \$XPSLINK from \$XPSPRE

Explanation: The volume name specified contains more than six characters.

System Action: Ends the operation.

User Response: Re-enter volume name with a maximum of 6 characters.

ERROR - WORK DATASET TOO SMALL (MINIMUM = 256 RECORDS)

Issued by: \$EDXLINK

Explanation: Length of work data set is less than 256 records.

System Action: Terminates \$EDXLINK and sets a completion code of 16.

User Response: Delete and reallocate the work data set to be at least 256 records.

ERROR - VOLUME NOT USABLE

Issued by: \$EDXLINK

Explanation: The volume requested is not usable.

System Action: Volume is not opened; \$EDXLINK continues, and sets a completion code of 8.

User Response: Determine why the volume is not usable and correct or use a different volume.

ERROR - WORK DATASET TOO SMALL. MINIMUM SIZE MUST xxx RECORDS

Issued by: \$XPSLINK from \$XPSPOST

Explanation: The work data set passed to \$XPSPOST from \$XPSLINK is not large enough to be used by \$XPSPOST.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Reallocate the work data set at the size specified (xxx) and restart \$XPSLINK.

ERROR WHILE TRYING TO READ CYCLE STEAL STATUS

Issued by: \$DASDI

Explanation: An attempt to print out the system cycle steal status failed.

System Action: Prompts for a retry.

User Response: Retry the function. If problem persists, the device is unusable.

**ERROR - WRITE ERROR ENCOUNTERED
ON xxx,yyy**

Issued by: \$EDXLINK

Explanation: A write error occurred while processing this data set.

System Action: \$EDXLINK will print the link map but no program will be stored. \$EDXLINK also sets a completion code of 12.

User Response: Allocate the data set in a different place.

ERROR WRITING HEADER RECORD xxx

Issued by: \$DASDI

Explanation: An error occurred while trying to write the header record on the device.

System Action: Ends the operation.

User Response: Retry the function. If problem persists the device is unusable.

ERROR WRITING LOG CONTROL RECORD

Issued by: \$LOG

Explanation: A write error occurred on the log data set.

System Action: Returns to command mode.

User Response: End \$LOG and use a different log data set or use \$DISKUT1 to allocate a new one. Then reload \$LOG with the new log data set.

**ERROR - \$XPSLINK/\$XPSPRE FOR
SUPERVISOR LINKS ONLY**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: You attempted to link a data set that was not the supervisor (\$EDXNUCx).

System Action: Ends the operation.

User Response: Link \$XPSLINK with the supervisor only. Use \$EDXLINK for all other links.

**ERROR: XPSTABLE ADDRESS IS
UNRESOLVED**

Issued by: \$XPSLINK from \$XPSPPOST

Explanation: \$XPSLINK did not resolve the XPSTABLE address. This is a fatal error to \$XPSPPOST.

System Action: \$XPSPPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Run \$XPSLINK again.

EVENT NAME IS NUMERIC - CAPRINT TERMINATED

Issued by: \$S1ASM

Explanation: The EVENT operand on the previous CAPRINT instruction is numeric. The first character must be alphabetic.

System Action: Terminates expansion of the CAPRINT instruction.

User Response: On the CAPRINT instruction, specify a valid event operand whose first character is alphabetic, and reassemble.

EVENT NAME TOO LONG - CAPRINT TERMINATED

Issued by: \$S1ASM

Explanation: The event operand on the previous CAPRINT instruction has more than eight characters.

System Action: Terminates expansion of the CAPRINT instruction.

User Response: On the CAPRINT instruction, specify an event operand that is less than or equal to eight characters in length, and reassemble.

EVENT NAME NOT SPECIFIED

Issued by: \$EDXASM

Explanation: For the RESET instruction, the event name (label of an ECB statement) must be specified. For process interrupt activity, use PIn, where "n" is a process interrupt number in the range of 1-99.

System Action: The error is flagged and compilation continues.

User Response: Specify the event name for RESET or PIn for process interrupt.

EXCEPTION - BLOCK CHECK ERROR

Issued by: \$BSCUT1

Explanation: During a trace of a BSC line, block checking detected an error in transmissions.

System Action: Formats the trace file up to the point of the error. Ends the operation.

User Response: Examine the trace file for the faulty transmission. You may want to retrace the line.

EXCEPTION - MODEM INTERFACE ERROR

Issued by: \$BSCUT1

Explanation: The trace of a BSC line shows that the modem or modem eliminator malfunctioned during transmissions.

System Action: Formats the trace file up to the point of the error. Ends the operation.

User Response: Check the connection between the BSC line and the modem or modem eliminator. Re-trace the line.

EXCEPTION - MULTIPOINT TRANSMIT ERROR

Issued by: \$BSCUT1

Explanation: During the trace of a BSC line defined for multipoint operations, the utility encountered an error in either receiving or sending data.

System Action: Formats the trace file up to the point of the error. Ends the operation.

User Response: Examine the trace file for evidence of the problem. You may need to check the BSC line to see that it is properly defined, attached, and supported by the system.

EXCEPTION - NOT DEFINED TO \$BSCUT1

Issued by: \$BSCUT1

Explanation: The device you specified for output of the trace file does not exist.

System Action: Ends the operation.

User Response: Specify an existing device to receive the output.

EXCEPTION - OVERRUN

Issued by: \$BSCUT1

Explanation: The size of the data set to hold the trace file was inadequate to contain all the trace records.

System Action: Ends the operation.

User Response: Retry the dump of trace data using a larger data set.

EXCEPTION - TIMEOUT

Issued by: \$BSCUT1

Explanation: During a trace of a BSC line, the system timed out while waiting to receive or send the next transmission.

System Action: Ends the operation.

User Response: Examine the trace file for evidence of the problem. You may want to change the timeout value for BSC line transmissions for your system.

E

EXCEPTION - WRONG LENGTH RECORD (xxx)

Issued by: \$BSCUT1

Explanation: The utility encountered a discrepancy between the length of one or more trace file records, and the length you specified for trace records with the \$BSCTRCE utility. The field xxx indicates that the record was either too short or too long.

System Action: Ends the operation.

User Response: Examine the trace file for evidence of the problem.

EXEC COMMAND NOT PRECEDED BY PROGRAM COMMAND

Issued by: \$JOBUTIL

Explanation: In the procedure data set, you have placed the elements in the wrong order.

System Action: Ends the utility.

User Response: Edit the procedure data set, placing the elements in the proper order. Then retry the utility.

EXPLICITLY DEFINED DECIMAL TERM INVALID FOR BYTE XP-MOVE

Issued by: \$EDXASM

Explanation: A decimal constant is not a valid immediate operand for a byte cross-partition move. If an immediate value is required, use hexadecimal (X) or character (C).

System Action: The error is flagged and compilation continues.

User Response: Recode the immediate value to be defined as hex or character data.

EXTENDED ERROR INFORMATION AVAILABLE, FUNCTION xxx - ERROR =yyy

Issued by: \$RJESNA

Explanation: When \$RJESNA issued the "NETWORK FUNCTION" error message, it determined that the SNA support had additional error information available.

The value for xxx indicates the base SNA or Event Driven Executive function that failed. The value for yyy indicates the return code received from that function.

System Action: Terminates \$RJESNA.

User Response: Use this message in conjunction with the "NETWORK FUNCTION" error message to find and correct the problem. \$RJESNA requires both native and host support. You may have to consult the programmer who wrote the host interface to help solve this problem. After the problem is corrected, restart \$RJESNA.

EXTRACT ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the PS command, an EXTRACT function was issued to \$IAM either to obtain the paging statistics or to reset the statistics. The EXTRACT failed, resulting in a return code of xxx.

System Action: If the failing EXTRACT was to obtain the statistics for display, the statistics are not displayed and the command terminates. If it was to reset the statistics, the statistics are not reset.

User Response: Check the Indexed Access Method return code to find the cause of the problem, and take appropriate action.

EXTRA STORAGE SPECIFIED AT LOAD TIME MUST BE 4608 BYTES OR GREATER

Issued by: \$IMAGE

Explanation: You loaded the utility and specified an amount of dynamic storage less than the required minimum.

System Action: \$IMAGE is not loaded.

User Response: Reload the utility and take the default storage amount, or specify storage greater than 4608 bytes.

EXTRACT FAILED FOR xxx,yyy

Issued by: \$IAMUT1

Explanation: An Indexed Access Method EXTRACT request was issued to the index file named in xxx,yyy.

System Action: Displays an Indexed Access Method return code. Ends the operation.

User Response: Examine the Indexed Access Method return code to find the cause of the problem, and take appropriate action.

FILE ALREADY IN USE

Issued by: \$IAMUT1

Explanation: Using the DF command, the indexed file currently being defined is already being used by another application.

System Action: DF command is terminated

User Response: Verify that the file name provided to DF is correct.

FILE SUBMISSION ABORTED

Issued by: \$RJESNA

Explanation: The current job stream submission has been terminated prior to completion. If you did not enter an ABORT command, it may be due to \$RJESNA's response to a host request (for example, the host system operator canceled the workstation's reader). The host may send a message specifying the reason for the abort.

It may also be due to a "concat" open error on a data set or to a read error. In these cases \$RJESNA has issued a previous message specifying the error.

System Action: Terminates the current job stream submission.

User Response: If \$RJESNA has issued a previous message, consult its explanation and correct the "concat" open error or read error. If the host issued a message, correct any problem that it indicates on your system (if possible) and retry the file submission.

FIRST BYTE OF BUFFER IS NOT ALPHABETIC

Issued by: \$S1ASM

Explanation: The first byte of the BUFFER= operand on the previous CAIOCB statement is numeric. The first byte must be alphabetic.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a valid BUFFER= operand whose first byte is alphabetic, and reassemble.

FIRST DIGIT OF ADDR FIELD NOT HEX

Issued by: \$S1ASM

Explanation: The device address operand on the previous CAxxx instruction has an invalid hexadecimal character as the first digit.

System Action: Terminates expansion of the CAPRINT instruction.

User Response: On the CAxxx instruction, specify a valid device address operand whose first digit is a hexadecimal character, and reassemble.

FIRST DIGIT OF PORT IS NOT NUMERIC

Issued by: \$S1ASM

Explanation: The PORT= operand on the CAIOCB statement has a nonnumeric character as the first digit.

System Action: Terminates the CAIOCB statement.

User Response: On the CAIOCB statement specify a valid PORT= operand whose first digit is numeric, and reassemble.

**FIXED HEAD INITIALIZATION ERROR,
DISK RETURN CODE = xxx IN THE TWO
RECORDS STARTING WITH RECORD
yyy**

Issued by: \$DASDI

Explanation: The read/write operation encountered an error during fixed head refresh (at IPL).

System Action: Fixed-head refresh continues for the rest of the fixed head area.

User Response: Assign an alternate sector on disk using \$DASDI. The verify option of \$INITDSK may aid you in this process.

**FLOATING POINT CONVERSION ERROR
OR EBFLCVT NOT IN SUPERVISOR**

Issued by: \$EDXASM

Explanation: An error was detected when trying to convert to binary the floating point or doubleword value specified on the DC or DATA statement.

System Action: The error is flagged and compilation continues.

User Response: Ensure floating point support is included in the supervisor and that the values to be converted are specified correctly.

**FORMAT - /-TYPE SPECIFICATION IS
INVALID**

Issued by: \$EDXASM

Explanation: An invalid slash (/) was found in the list operand structure of the FORMAT statement.

System Action: The error is flagged and compilation continues.

User Response: Ensure that slashes are used correctly in the list operand structure.

FORMAT - A-TYPE SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: The A-type specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid A-type specification for the list operand.

FORMAT - DELIMITER SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: An invalid delimiter was found in the list operand of the FORMAT statement.

System Action: The error is flagged and compilation continues.

User Response: Use a valid delimiter in the list operand.

FORMAT ERROR, CYLINDER xxx, HEAD yyy

Issued by: \$DASDI

Explanation: During initialization of a diskette on a 4966, \$DASDI was unable to format the diskette at cylinder xxx, head yyy.

System Action: \$DASDI prompts you for a retry.

User Response: Reply "Y" to the retry prompt. If error condition persists, remove that diskette and try another diskette.

FORMAT - EXCEEDS MAXIMUM NUMBER OF SPECIFICATIONS (80)

Issued by: \$EDXASM

Explanation: The number of consecutive constants specified on the FORMAT statement exceeds the maximum of 80.

System Action: The error is flagged and compilation continues.

User Response: Ensure the number of consecutive constants specified does not exceed 80.

FORMAT - F-TYPE SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: The F-type specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid F-type specification for the list operand.

FORMAT - H-TYPE SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: The H-type specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid H-type specification for the list operand.

FORMAT - I-TYPE SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: The I-type specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid I-type specification for the list operand.

FORMAT - MAXIMUM CHARACTER STRING IS 254

Issued by: \$EDXASM

Explanation: An error was detected in building a literal character string. The length of the character string exceeds the maximum of 254 bytes.

System Action: The error is flagged and compilation continues.

User Response: Ensure the literal character string does not exceed 254 characters in length.

FORMAT - NUMERIC SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: A numeric specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid numeric specification for the list operand.

FORMAT - X-TYPE SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: The X-type specification in the list operand of the FORMAT statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Code a valid X-type specification for the list operand.

FORMAT SPECIFICATION IS INVALID

Issued by: \$EDXASM

Explanation: An invalid conversion specification was coded in the list operand of the FORMAT statement.

System Action: The error is flagged and compilation continues.

User Response: Code a valid specification.

FORMAT SPECIFICATION NOT ALLOWED WITHIN GET/PUTEDIT

Issued by: \$EDXASM

Explanation: The (format,list) operand specification is invalid if the program is compiled under \$EDXASM; the "format" specification must, however, be present.

System Action: The error is flagged and compilation continues.

User Response: Specify the "format" operand specification only when using \$EDXASM.

F

FROM KEY SPECIFIED WITH #1.#2

Issued by: \$EDXASM

Explanation: Index register #1 or #2 cannot be specified as a source address (operand 2) in a cross-partition move.

System Action: The error is flagged and compilation continues.

User Response: Recode the cross-partition move without using an index register as the source address.

FROM KEY SPECIFIED WITH IMMEDIATE OPERAND

Issued by: \$EDXASM

Explanation: Operand 2, the From Key address, cannot be immediate data in a cross-partition move. The move instruction, in this case, moves data from an address in one partition to an address in another partition. Therefore, an address, and not immediate data, must be specified.

System Action: The error is flagged and compilation continues.

User Response: Recode the cross-partition move so that operand 2 is an address and not immediate data.

FUNCTION HAS COMPLETED WITH ERRORS. CHECK ALL OUTPUT

Issued by: \$MSGUT1

Explanation: The utility detected one or more errors during processing and sent error messages to the output terminal.

System Action: Returns to command mode.

User Response: Check the output listing for errors. Correct them and rerun the function.

FUNCTION IN PROCESS - PLEASE WAIT

Issued by: \$RJESNA

Explanation: The function you requested is currently processing another request.

System Action: \$RJESNA continues processing the current operation.

User Response: Wait for processing of the current operation to complete. Once the requested function becomes active, it is possible to abort it using the ABORT command.

FUNCTION xxx UNSUCCESSFUL. RC = yyy

Issued by: \$TAPEUT1

Explanation: The function you requested failed. The result was a tape read/write return code to find the cause of the problem, and take appropriate action.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**GET SEQUENTIAL TO PRIME DATASET
ERROR, INDEXED ACCESS METHOD
RTCODE = xxx**

Issued by: \$IAMUT1

Explanation: During the secondary index load, an Indexed Access Method error has occurred in the Sort/Merge input exit routine.

System Action: Sort/Merge terminates abnormally and you are prompted REPLY (Y/N)?

User Response: Check the Indexed Access Method return code to find the cause of the problem and reply accordingly.

**GPIB KEYBOARD TASK DOES NOT EXIST.
ATTENTION LISTS CANNOT BE DEFINED.
CANNOT CHANGE PARTITION
NUMBER**

Issued by: \$GPIBUT1

Explanation: No keyboard task is defined for the device you are using.

System Action: Ends the operation.

User Response: Define a keyboard task for the device and retry the function.

GPIB NOT ATTACHED

Issued by: \$GPIBUT1

Explanation: The General Purpose Interface Bus is not physically attached to your system. It will not function and you cannot use the GPIB utility.

System Action: Ends the operation.

User Response: If your system is supposed to have the GPIB capability, check the installation of the bus feature to make sure it is properly connected, defined and supported by your system.

**GPIB TERMINAL NOT DEFINED TO
UTILITY**

Issued by: \$GPIBUT1

Explanation: You attempted to use a device that the utility does not recognize.

System Action: Ends the operation.

User Response: Define the terminal to the utility and retry the function.

HEADER READ ERROR

Issued by: \$PREFIND

Explanation: A disk error occurred while reading a program header.

System Action: Returns to command mode.

User Response: Recreate the program and rerun the function.

HIGH KEY IN RBN xxx DOES NOT MATCH INDEX ENTRY IN RBN

Issued by: \$VERIFY

Explanation: The index block which points to the block xxx shows a discrepancy between the designated high key and the actual high key for that block.

System Action: Verification continues.

User Response: File is invalid and will have to be re-created. Contact the person who created the file for this purpose.

HOST LINK DOWN TP ERROR: RC = xxx

Issued by: \$UPDATEH

Explanation: The connection with the host processor is inoperative. Processing cannot continue. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the Host Communication Facility TP return code to find the cause of the problem, and take appropriate action. You will have to wait until the link is restored before you can retry.

HOST REJECTED CONNECTION

Issued by: \$RJESNA

Explanation: The host access method (VTAM or TCAM) or the host entry system has rejected the logon request from \$RJESNA.

System Action: Terminates \$RJESNA.

User Response: Verify the following:

1. The logon parameters you entered are correct.
2. The host entry system is prepared to accept the logon request (such as logical line active).
3. The session parameters at the host are correct. For example, VTAM mode table name.

\$RJESNA requires native and host support. You may have to consult the programmer who wrote the host interface to help solve this problem. After the problem is corrected, restart \$RJESNA.

HOST/SEND PGM ERROR - REMOTE RECORD NOT STORED

Issued by: \$LOG

Explanation: The \$LOG utility cannot find the host program (CJUALTHL) or the send program (CJUALTSL) for the Remote Manager. These programs must be loaded before the Remote Manager can receive error log records from \$LOG.

System Action: \$LOG is active but it cannot write log records to the Remote Manager's buffers.

User Response: Ensure that the host program and the send program are loaded.

**IA BUFFER LENGTH NOT BETWEEN 10
AND 100**

Issued by: \$EDXASM

Explanation: The number specified for the maximum number of buffered interrupts is not within the valid range. The value for the IABUF= operand of the SYSTEM statement must be a positive integer in the range 10–100.

System Action: The error is flagged and compilation continues.

User Response: Specify a number from 10 - 100 for the IABUF= operand.

**\$IAM HAS INCURRED A SEVERE ERROR
\$IAM CENTRAL BUFFER ADDRESS IS
xxx - ADDRESS SPACE yyy**

Issued by: \$IAM

Explanation: \$IAM program storage has been overwritten. Indexed data set control information has been destroyed.

System Action: \$IAM goes into an unrecoverable wait and will not process any access requests.

User Response: You can dump the central buffer with the \$D system command and take appropriate action to stop your application.

You can use the recovery and backup procedures to restore the data set, or you can resume execution of your application. To restart your application, you can either IPL or cancel \$IAM and reload it.

**\$IAM PROGRAM NOT FOUND ON EITHER
THE IPL VOLUME OR VOLUME xxx**

Issued by: \$IAMUT1

Explanation: The program was attempting to read the \$IAM program to obtain current storage information. The read was attempted first for the IPL volume, then for the volume containing the \$IAMUT1 program. Neither volume contained the \$IAM program.

System Action: Ends the operation.

User Response: Install \$IAM on either the IPL volume or on the volume containing the \$IAMUT1 program.

\$IAMQCB IS NOT INCLUDED IN SYSTEM

Issued by: \$IAMUT1

Explanation: In order to access the directory data set (DR command), \$IAMQCB must be included in the system.

System Action: Ends the operation.

User Response: Include the \$IAMQCB in the system generation and retry.

\$IAM SHUTTING DOWN. INSUFFICIENT DYNAMIC STORAGE

Issued by: \$IAM

Explanation: The amount of storage specified for the \$IAM module was not sufficient for control blocks and buffers. Either the storage parameter was specified where \$IAM was loaded, or the amount of storage was changed via the SS command of \$DISKUT2. Neither of these should be done.

System Action: \$IAM terminates.

User Response: Issue the BF command of \$IAMUT1. This will reset the storage to an amount consistent with the number of control blocks and buffers.

IBIT MUST BE EITHER 0 OR 1

Issued by: \$EDXASM

Explanation: The setting of IBIT= operand determines whether a device can present an interrupt when the PREPARE command of the IDCB instruction is issued.

System Action: The error is flagged and compilation continues.

User Response: Specify either IBIT=OFF (indicates 0) or IBIT=ON (indicates 1).

IDCBADDR POSITIONAL PARAMETER MISSING

Issued by: \$EDXASM

Explanation: The "idcbaddr" operand of the EXIO instruction is required and must specify the address of the IDCB.

System Action: The error is flagged and compilation continues.

User Response: Specify the IDCB address for the "idcbaddr" operand.

ID= MUST BE SPECIFIED, AND A UNIQUE 1 - 6 CHARACTER LABEL

Issued by: \$EDXASM

Explanation: The ID= operand of the TAPE statement is required and must specify a 1 - 6 character name that will be associated with the device.

System Action: The error is flagged and compilation continues.

User Response: Specify a 1 - 6 character name for the ID= operand.

IF/DO NESTING LIMIT EXCEEDED

Issued by: \$EDXASM

Explanation: The limit of 20 nested loops was exceeded.

System Action: The error is flagged and compilation continues.

User Response: Reduce the number of nested loops.

IF WAIT=NO, ERROR= AND END= MAY NOT BE SPECIFIED

Issued by: \$EDXASM

Explanation: If WAIT=NO is specified on the READ/WRITE instruction, you cannot code the ERROR= and END= operands. Any error, completion, or end-of-file checking must be performed by the application program.

System Action: The error is flagged and compilation continues.

User Response: Do not specify ERROR= or END= if WAIT=NO is coded.

ILLEGAL CONTINUATION - DATA MUST START IN COLUMN 16

Issued by: \$EDXASM

Explanation: All continuations of a line of code must begin in column 16. The continuation character must be coded in column 72.

System Action: The error is flagged and compilation continues.

User Response: Begin the continuing code in column 16 with column 72 containing the continuation character; otherwise remove the continuation character.

ILLEGAL DISKETTE - NO VOL1 LABEL

Issued by: \$INITDSK

Explanation: You tried to perform a function on a diskette that was not formatted for use by EDX.

System Action: Returns to command mode.

User Response: Use a diskette formatted for EDX or use \$DASDI to reinitialize the original diskette.

ILLEGAL INPUT RECORD FORMAT

Issued by: \$UPDATE,\$UPDATEH

Explanation: You specified a data set that is not an object module and therefore cannot be built into a program.

System Action: Returns to command mode.

User Response: Specify a data set that contains an object module, or create an object module out of the original data set.

ILLEGAL NUMBER OF OPERANDS - MUST BE EVEN

Issued by: \$EDXASM

Explanation: An odd number of operands for the attention interrupt handling routine was specified on the ATTNLIST instruction.

System Action: The error is flagged and compilation continues.

User Response: An even number of operands must be specified on the ATTNLIST instruction. Two entries are needed to specify each attention interrupt handling routine.

ILLEGAL - ODD ADDRESS

Issued by: \$STGUT1

Explanation: You specified an odd address and the utility can only accept an even address.

System Action: Returns to command mode.

User Response: Retry the function and specify a correct address, or re-IPL the system.

INADEQUATE STORAGE, MIN BYTES REQUIRED

Issued by: \$IAMUT1

Explanation: Using the LO, UN, or RO commands, the minimum storage required by the requested function is not available.

System Action: Ends the operation.

User Response: Increase the amount of available storage and retry the function.

INCOMPATIBLE MARGINS

Issued by: \$EDXASM

Explanation: When specifying the format of the output medium, the problem may be one of the following:

- Right margin specified is not between left margin and LINSIZE-1
- Left margin is not between zero and LINSIZE-1

System Action: The error is flagged and compilation continues.

User Response: Specify the correct margin dimensions as explained.

INCORRECT DEVICE TYPE - MUST BE AN IBM DISK OR DISKETTE

Issued by: \$DASDI

Explanation: The device at the address entered in the INIT operator command is not 4962 disk, 4963 disk, 4967 disk, or 4964, 4965, or 4966 diskette.

System Action: The INIT function is terminated.

User Response: Specify the address of an IBM disk or diskette in another INIT command.

**INCORRECT PORTS SPECIFICATION -
EXPANSION TERMINATED**

Issued by: \$EDXASM

Explanation: The PORTS= operand on the CACBx statement is incorrectly specified.

System Action: Ends the expansion.

User Response: Code a decimal number between zero and 31 for the PORTS= operand.

**INCORRECT WRAP COUNT SPECIFIED
LOG DATA SET WRAP COUNT IS
xxx HEX**

Issued by: \$DISKUT2

Explanation: You entered an incorrect wrap count while using the PR or LR command of the \$DISKUT2 utility.

System Action: The system asks you for the correct wrap count.

User Response: Enter the wrap count displayed in the message.

**INCREASE BUFFER SIZE BEFORE
REATTEMPTING OPERATION**

Issued by: \$IAMUT1

Explanation: Using the LO, UN, RO commands, the minimum storage required by the requested function is not available.

System Action: Ends the operation.

User Response: Increase the amount of available storage and retry the function.

**INDEX ACCESS METHOD READ/WRITE
ERROR. CODE=xxx**

Issued by: \$VERIFY

Explanation: While performing verification of secondary indexes, an Indexed Access Method I/O error was encountered.

System Action: \$VERIFY terminates.

User Response: Check the Indexed Access Method return code to find the cause of the problem, and take appropriate action.

**INDEX INCORRECTLY SPECIFIED - CAxxx
TERMINATED**

Issued by: \$S1ASM

Explanation: An index field contains an entry other than #1, #2; or more than two fields are specified.

System Action: Terminates expansion of the CAPRINT instruction.

User Response: Specify #1 or #2 in the index field and make sure the index operand has only two fields.

INDEXING NOT ALLOWED FOR THIS INSTRUCTION

Issued by: \$EDXASM

Explanation: You coded an indexed address (parameter,#register) in an instruction that does not allow indexing.

System Action: The error is flagged and compilation continues.

User Response: Code an address instead of an indexed address for the instruction operand and recompile the program.

INDEX REGISTER IS AN INVALID OPERAND

Issued by: \$EDXASM

Explanation: The use of an index register as opnd1 or opnd2 is invalid for this instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify opnd1 or opnd2 as explained in the instruction description.

INITIALIZATION ERROR

Issued by: \$MTM

Explanation: Initialization has been unsuccessful. This message is written to the terminal which loaded the Multiple Terminal Manager. Additional messages are printed on the Multiple Terminal Manager log device.

System Action: Terminates Multiple Terminal Manager.

User Response: Based on the information in the additional messages that print out on the log device, determine the cause of the error and correct it.

INITIALIZATION WORK AREA OVERFLOW

Issued by: \$MTM

Explanation: The work area overflowed for the CDMINIT2 initialization routine. As distributed, the CDMINIT2 routine has work area space for up to 36 terminals.

System Action: The initialization process terminates and no Communication Facility stations are started.

User Response: Reduce the number of terminals defined in the STATIONS file.

IN ORDER TO RELOCATE THE VOLUME DIRECTORY IN THE DEVICE, THE NEW RECORD NUMBER MUST BE GREATER THAN THE DEFAULT

Issued by: \$INITDSK

Explanation: You entered a location for the device directory which is lower than the default location.

System Action: Returns to command mode.

User Response: Retry the function with a location higher than the default, or specify the default.

INPUT DATA SET: xxx NOT FOUND

Issued by: \$UPDATE

Explanation: The name you specified as the object module does not exist.

System Action: Returns to command mode.

User Response: Specify the correct name for an existing object module.

**INPUT REC GT OUTPUT REC.
TRUNCATION WILL OCCUR**

Issued by: \$IAMUT1

Explanation: Using the LO, UN, or RO command, the input record size is larger than the output record size. The records must be truncated.

System Action: Issues prompt "OK TO PROCEED?"

User Response: Reply to the prompt accordingly.

INPUT RECORD ASSUMED TO BE xxx BYTES. OK?

Issued by: \$IAMTU1

Explanation: Using the LO command, the input record size is assumed to be the same as the output record size.

System Action: Awaits operator response.

User Response: Reply "Y" to continue processing with the indicated record size. Reply "N" to receive a prompt for the actual record size.

**INSUFFICIENT BUFFERS FOR LOG RATE;
NUMBER OF TIMES: xxx**

Issued by: \$LOG

Explanation: The \$LOG buffers cannot handle the number of I/O error interrupts that are occurring. The number of I/O error interrupts that were lost when the buffers overflowed is xxx.

System Action: None.

User Response: Print the contents of the log data set with the \$DISKUT2 utility to find the source of the I/O error interrupts.

INSUFFICIENT CONTIGUOUS SPACE

Issued by: \$INITDSK

Explanation: The requested function requires more contiguous space than exists on the volume in use.

System Action: Returns to command mode.

User Response: Compress the volume to provide more contiguous space. Then retry the \$INITDSK function. Or, allocate a smaller volume and retry the function.

INSUFFICIENT CONTIGUOUS SPACE AT END OF VOLUME

Issued by: \$INITDSK

Explanation: You tried to split a volume and it didn't have enough space at the end.

System Action: Ends the operation.

User Response: Compress the volume to provide more contiguous space, then retry. Or, retry the function and allocate a smaller volume.

INSUFFICIENT STORAGE

Issued by: \$HXUT1

Explanation: \$HXUT1 attempted to load \$DISKUT3 and sufficient storage was unavailable in the partition to successfully load \$DISKUT3.

System Action: The command terminates.

User Response: Cancel some of the other programs executing (at your discretion) to free a portion of storage large enough to contain \$DISKUT3 and retry.

INSUFFICIENT STORAGE AVAILABLE FOR TERMINAL PROCESSING

Issued by: \$EDXASM

Explanation: The terminal device handler was unable to load the program for the device; insufficient storage in the partition.

System Action: The error is flagged and compilation continues.

User Response: Load \$EDXASM in the partition by itself or load \$EDXASM in a larger partition.

INSUFFICIENT STORAGE AVAILABLE, 512 BYTES REQUIRED

Issued by: \$IAMUT1

Explanation: Using the DR command, the directory function was not able to obtain a 512 byte buffer.

System Action: Ends the operation.

User Response: Increase the amount of dynamic storage and try again.

INSUFFICIENT STORAGE FOR DATA STREAM BUFFER

Issued by: Spooling

Explanation: The spool writer was unable to obtain a work area to contain the spool job's data stream

System Action: Ends the operation.

User Response: Free storage in the partition in use and restart the writer.

INSUFFICIENT STORAGE FOR WRITER

Issued by: Spooling

Explanation: An attempt was made to start a writer when not enough main storage space was available.

System Action: Does not start the spool writer.

User Response: Use the \$A command to display the programs in the partition. Obtain adequate space in the partition to start the writer by either:

- Cancelling an unneeded program and/or restarting it in another partition.
- Waiting until a program terminates.

INSUFFICIENT STORAGE TO ALLOCATE TARGET DATA SET - DATA SET NOT ALLOCATED

Issued by: \$HXUT1

Explanation: The data set allocation operation by \$DISKUT3 failed because of insufficient storage on the target volume.

System Action: The command terminates.

User Response: Specify a different volume for the target data set, or free a portion of storage in the target volume (DE command of \$DISKUT1 utility) and retry.

INTERFACE DATA CHECK

Issued by: \$GPIBUT1

Explanation: The system is checking for data errors.

System Action: Continues processing.

User Response: Investigate any errors, correct them, and retry the function.

INTERNAL 3101 DATA STREAM TOO LARGE. 3101 ATTR CHARACTERS WILL NOT BE USED

Issued by: \$IMAGE

Explanation: The utility will not use the supplied data stream to build a screen image.

System Action: Prompts for the enter key.

User Response: Either save the screen without attribute bytes or reduce content of the screen.

INVALID ADDRESS=, MUST BE BETWEEN '00' AND 'FF'

Issued by: \$EDXASM

Explanation: The address must be specified and be within the range X'00' to X'FF'.

System Action: The error is flagged and compilation continues.

User Response: Specify the address between the valid range.

INVALID ADDRESS OPERAND

Issued by: \$EDXASM

Explanation: The address specified for the ADDRESS= operand of the BSCLINE statement is not in the range of X'00' to X'FF'.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid address.

INVALID ADDRESS - STORAGE NOT RELEASED

Issued by: \$STGUT1

Explanation: The address you specified is not valid for the function you requested.

System Action: Returns to command mode.

User Response: To find the correct address, issue the \$A command. Then retry the function using the correct address.

INVALID AITYPE=

Issued by: \$EDXASM

Explanation: An invalid entry was coded for AITYPE with a 4982 sensor I/O device specified.

System Action: The error is flagged and compilation continues.

User Response: Recode the AITYPE operand in the SENSORIO statement using the valid entries:

- RR or RELAY for relay
- SS or SOLID for solid state

INVALID AUTO-START

Issued by: \$MTM

Explanation: The program name specified in the TERMINAL file to be auto-started was invalid or auto-start was specified with signon for a real terminal.

System Action: The specified program is not auto started on the terminal.

User Response: Specify a valid name for the program (in TERMINAL file) to be auto-started or specify no signon for a real terminal.

INVALID BIT=, MUST BE BETWEEN 0 AND 15

Issued by: \$EDXASM

Explanation: The BIT= operand must be specified and be within the range of 0–15 for PI sensor-based IODEF instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the bit range within 0–15.

INVALID BIT RATE/RANGE SPECIFICATION

Issued by: \$EDXASM

Explanation: The combination of BITRATE and RANGE operands specified on this TERMINAL statement are not for this device.

System Action: The error is flagged and compilation continues.

User Response: Refer to the TERMINAL statement in the *Installation and System Generation Guide* for the appropriate specifications for these operands for this device.

INVALID BITRATE

Issued by: \$EDXASM

Explanation: You specified a line speed other than 100, 250, or 500 on the TERMINAL statement for the 4980 display station.

System Action: The error is flagged and compilation continues.

User Response: Specify a line speed of 100, 250, or 500 for the BITRATE operand of the TERMINAL statement.

INVALID BIT RATE SPECIFIED

Issued by: \$EDXASM

Explanation: If RANGE=HIGH is coded, the value of the BITRATE operand must be at least 300 bits per second. If RANGE=LOW is coded, the value of the BITRATE operand must not exceed 1200 bits per second. When ADAPTER=MFA is specified, the BITRATE value must be 1200, 2400, 4800, or 9600 bits per second.

System Action: The error is flagged and compilation continues.

User Response: Code a valid bit rate in the TERMINAL statement.

INVALID BIT RATE FOR DEVICE SPECIFIED

Issued by: \$EDXASM

Explanation: The bit rate for the 4975-01R must be 1200 bps. For a 4975-02R, the bit rate must be 1200, 2400, or 4800 bps.

System Action: The error is flagged and compilation continues.

User Response: Specify the the correct bit rate. If BITRATE= is omitted, 1200 bps is the default.

INVALID BITS=, MUST HAVE THE FORM BITS=(U,V)

Issued by: \$EDXASM

Explanation: The BITS= operand of the SBIO instruction must have the format BITS=(U,V), where U is the starting bit number and V is the length of the bit string.

System Action: The error is flagged and compilation continues.

User Response: Specify the starting bit number and bit length as explained.

INVALID BOTTOM MARGIN

Issued by: \$EDXASM

Explanation: The syntax for the BOTM= operand of the TERMINAL statement was specified incorrectly. The value must be a positive self-defining term. The value must be between TOPM+NHIST and PAGESIZE-1.

System Action: The error is flagged and compilation continues.

User Response: Specify a positive self-defining term for the BOTM= operand.

INVALID BSC LINE ADDRESS

Issued by: \$BSCTRCE

Explanation: You specified a BSC line that the system cannot trace, for one of these reasons: The line is not attached to the system; the line is not supported by the supervisor; or the line is not currently in use by an executing BSC program.

System Action: Awaits operator response.

User Response: Determine the addresses of valid BSC lines by using the \$IOTEST utility. If possible, check the source code of the executing BSC program to determine which line it uses. Then retry the trace specifying a valid BSC line address.

INVALID BSCIOCB ADDRESS SPECIFICATION

Issued by: \$EDXASM

Explanation: The address specified on the BSCIOCB statement is not within the valid range of X'00' to X'FF' (hexadecimal).

System Action: The error is flagged and compilation continues.

User Response: Specify an address within the valid range.

INVALID BSCREAD/BSCWRITE TYPE SPECIFICATION

Issued by: \$EDXASM

Explanation: The operation specified for “type” on the BSCREAD, BSCWRITE, or BSCLINE statements is invalid.

System Action: The error is flagged and compilation continues.

User Response: Check the valid types that can be specified.

INVALID 'BUSY' SPECIFICATION

Issued by: \$EDXASM

Explanation: The BUSY= operand on the ENQ instruction must be a label specifying the address of the routine to execute when the busy condition (resource unavailable) is encountered.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid label (1–8 characters) for the BUSY= operand.

INVALID CHARSET= KEYWORD SPECIFIED ON 4975 TERMINAL STATEMENT

Issued by: \$EDXASM

Explanation: The character set specified for the 4975 printer is not one of the available character sets or was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Refer to the description of the TERMINAL statement for the valid CHARSET= operands.

INVALID 'COMMAND='

Issued by: \$EDXASM

Explanation: An invalid I/O operation was specified in the COMMAND operand of an IDCB instruction.

System Action: The error is flagged and compilation continues.

User Response: Code a valid I/O operation.

INVALID COMMAND

Issued by: Any Utility

Explanation: The command you entered is not one of the valid commands for the utility in use.

System Action: Depending on the utility, either prompts for a new command, or waits for the operator to press the attention key and enter a command.

User Response: To obtain a list of the valid commands for the utility, enter a ? in response to the COMMAND prompt, or enter HELP, depending on the utility. Then enter a valid command.

INVALID COMMON= SPECIFICATION

Issued by: \$EDXASM

Explanation: During system generation, you coded the COMMON operand on the SYSTEM statement incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Refer to the SYSTEM statement description in the *Installation and System Generation Guide* for details on how to code the COMMON= operand. If you code more than one parameter for this operand, you must enclose the parameter list in parentheses. If you code only one parameter for the operand, do *not* enclose that parameter in parentheses.

INVALID COMP= SPECIFICATION

Issued by: \$EDXASM

Explanation: The COMP= operand must be a label of a COMP instruction which will indicate the four character ID of the message data set or storage resident messages.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid label for the COMP= operand.

INVALID CONJUNCTION SPECIFIED (MUST BE 'AND' OR 'OR')

Issued by: \$EDXASM

Explanation: Neither "AND" nor "OR" was specified.

System Action: The error is flagged and compilation continues.

User Response: Ensure that only an "AND" or an "OR" conjunction is specified.

INVALID COPYCODE LIBRARY NAME

Issued by: \$EDXASM

Explanation: The copy code volume listed on the *COPYCOD statement cannot be found.

System Action: The error is flagged and compilation continues.

User Response: Examine the name of the copy code volume on the *COPYCOD statement and correct any errors. The statement is contained within the language control data set.

INVALID COUNT= PARAMETER

Issued by: \$EDXASM

Explanation: The value specified for COUNT= on the FPCONV instruction must be a positive self-defining term.

System Action: The error is flagged and compilation continues.

User Response: Specify a positive self-defining term for the number of values to be converted on the COUNT= operand.

INVALID DATA IN DIRECTORY CONTROL ENTRY.

Issued by: \$COMPRES

Explanation: Invalid data was found in the directory control area. The directory control is damaged.

System Action: Prompts for another compress.

User Response: Reinitialize and restore the volume or device being compressed.

INVALID DATA TYPE COMBINATION

Issued by: \$EDXASM

Explanation: This error is associated with the SCREEN instruction in the graphics package. The screen coordinates entered were not the same data type.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the screen coordinates are the same data type.

INVALID DATA TYPE SPECIFIED

Issued by: \$EDXASM

Explanation: The data type (such as byte, word) specified for this instruction is invalid.

System Action: The error is flagged and compilation continues.

User Response: Ensure the required data type is properly specified for this instruction.

INVALID DATE FORMAT

Issued by: \$EDXASM

Explanation: The DATEFMT operand on the SYSTEM statement is specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the format as either "MMDDYY" or "DDMMYY". If omitted, "MMDDYY" is the default.

INVALID DCE ENCOUNTERED

Issued by: \$DIRECT

Explanation: Invalid data was found in the directory record.

System Action: Ends the operation.

User Response: Delete the data set and re-create it.

INVALID DENSITY

Issued by: \$TAPEUT1

Explanation: The density you specified is wrong for the type of tape in use.

System Action: Returns to command mode.

User Response: Specify a valid density for the tape in use.

INVALID DEVICE ADDRESS

Issued by: \$TERMUT1

Explanation: A device doesn't exist at the address you specified.

System Action: Returns to command mode.

User Response: Determine the valid device addresses with \$IOTEST and retry the function.

INVALID DEVICE ADDRESS, MUST BE OMITTED FOR 4013

Issued by: \$EDXASM

Explanation: An address must not be specified for a 4013 graphics terminal.

System Action: The error is flagged and compilation continues.

User Response: Delete the address.

INVALID DISKETTE MOUNTED MUST BE xxx

Issued by: \$MOVEVOL

Explanation: The diskette mounted is invalid for use by the utility.

System Action: Prompts for another diskette.

User Response: Mount a valid diskette type and retry the function.

INVALID 'DS=' ON LOAD

Issued by: \$EDXASM

Explanation: The data set specification on the LOAD instruction is coded incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Refer to the description of the DS operand in the LOAD instruction for the various ways to specify data sets.

INVALID 'DS=' ON OVERLAY LOAD, MUST HAVE THE FORM 'DSX'

Issued by: \$EDXASM

Explanation: The data set specified for overlay load was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the DS= operand in the form:

DS=(DS1,DS2,DS3,...,DS9)

INVALID DS= SPECIFICATION

Issued by: \$EDXASM

Explanation: An error in the DS= operand was detected while trying to construct the DSCB data for the PROGRAM instruction.

System Action: The error is flagged and compilation continues.

User Response: Ensure the DS= operand is coded correctly.

INVALID 'EOB' SPECIFIED

Issued by: \$EDXASM

Explanation: The label specified for EOB= on the SBIO instruction was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the label (1–8 characters) of the instruction to be executed upon completion of buffer operations with automatic indexing.

INVALID DUPLICATION FACTOR

Issued by: \$EDXASM

Explanation: In an OTE instruction, the operand DUPFAC (the duplication factor for the object text element) must be a self-defining term.

System Action: The error is flagged and compilation continues.

User Response: Recode the DUPFAC operand with a self-defining term.

INVALID 'END=', MUST BE 'YES' OR 'NO'

Issued by: \$EDXASM

Explanation: The valid parameters for the END= operand are "YES" and "NO". The *last* device definition statement within that device type group must have END=YES specified. END=NO can be specified or defaulted to for all other device definition statements of that group.

System Action: The error is flagged and compilation continues.

User Response: Specify either END=YES or END=NO as appropriate.

INVALID 'ERROR=' SPECIFIED

Issued by: \$EDXASM

Explanation: The optional ERROR= operand on the SBIO instruction must be a label specifying the address of the routine to execute when the error condition is encountered.

System Action: The error is flagged and compilation continues.

User Response: Specify the label (1–8 characters) of the routine on the ERROR= operand.

INVALID FORMAT= SPECIFICATION

Issued by: \$EDXASM

Explanation: The FORMAT= operand on the CONVTB/CONVTD instruction is specified incorrectly. The syntax must be of the form FORMAT=(W,D,T) and all fields must be specified.

System Action: The error is flagged and compilation continues.

User Response: Refer to the *Language Reference* for details on this operand.

INVALID FIRST CHARACTER IN PREC=

Issued by: \$EDXASM

Explanation: The first character in the PREC= specification must be either F (single precision), L (extended precision), or * (single precision as the default) for floating-point arithmetic operations.

System Action: The error is flagged and compilation continues.

User Response: Specify the first character for PREC= as explained.

INVALID FORMAT IN OPERAND 1

Issued by: \$EDXASM

Explanation: The format of operand 1 on the FIND instruction must be a character specified as C'x' or as a bit string specified as X'xx'.

System Action: The error is flagged and compilation continues.

User Response: Specify operand 1 as a character or hexadecimal byte.

INVALID FORMAT= SPECIFICATION

Issued by: \$EDXASM

Explanation: The width field of the FORMAT= operand is specified incorrectly on the GETVALUE or PRINTNUM instruction.

System Action: The error is flagged and compilation continues.

User Response: For the width, specify a decimal value equal to the maximum field width (in bytes) expected from the terminal.

INVALID GPIB OPERATION

Issued by: \$EDXASM

Explanation: An invalid command or option was requested on the TERMCTRL GPIB statement.

System Action: The error is flagged and compilation continues.

User Response: Refer to the description of the TERMCTRL instruction for GPIB for the valid operations.

INVALID HEXADECIMAL CONSTANT SPECIFIED

Issued by: \$EDXASM

Explanation: The system encountered a hexadecimal constant that was not written in proper hex notation. Valid hexadecimal characters are 0-9,A-F.

System Action: The error is flagged and compilation continues.

User Response: Recode the data statement or instruction using valid hexadecimal characters.

INVALID HEXADECIMAL ENTRY

Issued by: \$EDXASM

Explanation: An invalid hexadecimal character was detected. Valid hexadecimal characters are 0-9 and A-F.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid hexadecimal entry and reassemble.

INVALID IMMEDIATE OPERAND IN STRING COMPARE

Issued by: \$EDXASM

Explanation: In a string compare, opnd1 or opnd2 cannot be immediate data.

System Action: The error is flagged and compilation continues.

User Response: You can change the immediate data to a label and specify the string in a DATA statement.

INVALID INDEX REGISTER SPECIFICATION - NOT #1 OR #2

Issued by: \$EDXASM

Explanation: Only index registers #1 and #2 are valid.

System Action: The error is flagged and compilation continues.

User Response: Correct the error by specifying either #1 or #2.

INVALID 'INITPRT' PARM ON 'SYSTEM'

Issued by: \$EDXASM

Explanation: The partition number designated for initialization modules to execute in must be within the range of the value specified on the PARTS= operand (number of partitions) of the SYSTEM statement.

System Action: The error is flagged and compilation continues.

User Response: Specify a valid partition number (determined by PARTS=) for the INITPRT= operand.

INVALID IODA

Issued by: \$TAPEUT1

Explanation: The device address you specified is invalid.

System Action: Ends the operation.

User Response: Specify the correct address of an existing device.

INVALID KEYWORD COMBINATION

Issued by: \$EDXASM

Explanation: No other operand can be coded on the TERMCTRL instruction if the DCB= operand is coded.

System Action: The error is flagged and compilation continues.

User Response: Ensure no other operands are specified with the DCB= operand.

INVALID LENGTH= SPECIFICATION

Issued by: \$EDXASM

Explanation: The length specified must be a self-defining term not exceeding 254.

System Action: The error is flagged and compilation continues.

User Response: Specify the length as a self-defining term not exceeding 254.

INVALID LENGTH SPECIFIED ON LINE #xxx

Issued by: \$MSGUT1

Explanation: You supplied the wrong length for the type of operand specified.

System Action: Continues processing.

User Response: When processing finishes, correct the error and retry.

INVALID LEVEL SPECIFICATION

Issued by: \$EDXASM

Explanation: The number specified for the hardware interrupt level on the SENSORIO statement is invalid.

System Action: The error is flagged and compilation continues.

User Response: Specify a number between 0–3 as the interrupt level.

INVALID LINSIZE - MUST BE LESS THAN 255

Issued by: \$EDXASM

Explanation: The LINSIZE= operand of the TERMINAL statement must not exceed 254.

System Action: The error is flagged and compilation continues.

User Response: Specify a value for the LINSIZE= operand not exceeding 254.

INVALID 'LOGMSG=', MUST BE 'YES' OR 'NO'

Issued by: \$EDXASM

Explanation: The parameter specified for the LOGMSG= operand on the LOAD instruction was coded incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify "YES" if the "program loaded" message is to print; otherwise specify "NO".

INVALID 'LSB' SPECIFIED

Issued by: \$EDXASM

Explanation: The LSB= operand is valid only when the BITS= operand is coded on the SBIO instruction. Also an error can result if the range specification on the BITS= operand is equal to or greater than the value specified for LSB=.

System Action: The error is flagged and compilation continues.

User Response: Ensure the BITS= operand is coded and that the bit range is less than the value for LSB.

INVALID MC OPERAND

Issued by: \$EDXASM

Explanation: The multi-line controller operand (MC=) is invalid.

System Action: The error is flagged and compilation continues.

User Response: Specify MC=YES if multi-line controller; otherwise, MC=NO for single line.

INVALID, MISSING, OR DUPLICATE ADAPTER ADDRESS

Issued by: \$EDXASM

Explanation: The adapter address must be specified on the ADAPTER statement. A valid adapter address is a 2-digit hexadecimal number which must be divisible by 4.

System Action: The error is flagged and compilation continues.

User Response: Ensure the adapter address specified is coded only once and that it is specified correctly.

INVALID 'MODE=', MUST BE 'FIELD', 'LINE', OR 'SCREEN'

Issued by: \$EDXASM

Explanation: An invalid operand was specified for the MODE= operand of the ERASE instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify either "FIELD", "LINE", or "SCREEN" as the operand for MODE=.

INVALID MODE SPECIFICATION

Issued by: \$EDXASM

Explanation: The MODE= operand was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Refer to the *Language Reference* for the valid responses to this operand on the instruction being used.

INVALID NAME

Issued by: \$TAPEUT1

Explanation: You entered a data set name more than 8 characters long.

System Action: Returns to command mode.

User Response: Reenter valid data set name.

INVALID NAME

Issued by: \$XPSLINK from \$XPSPOST

Explanation: In response to a prompt for an output program data set, the name you entered was too long (more than 8 characters).

System Action: Reprompts for an output program data set name.

User Response: Reenter a valid data set name.

INVALID NAME HAS BEEN POSITIONED

Issued by: \$DIRECT

Explanation: During the sort process, an invalid name has been positioned.

System Action: Continues processing.

User Response: When the process is done, delete the invalid name that you have tried to position.

INVALID NO. OF ELEMENTS IN OPERAND - SHOULD BE 1 OR 2

Issued by: \$EDXASM

Explanation: The number of elements specified in one of the operands is invalid for the instruction.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the number of elements you specified is correct for that operand.

INVALID NUCLEUS NAME

Issued by: \$INITDSK

Explanation: The name you specified does not refer to an actual nucleus.

System Action: Ends the operation.

User Response: Specify a name for a valid nucleus.

**INVALID NUMBER; MUST BE IN RANGE
OF 1 TO xxx**

Issued by: \$COPYUT1

Explanation: The system found an invalid length in the member control record. The member directory entry is damaged.

System Action: Ends the operation.

User Response: Recreate the member directory entry.

INVALID NUMERIC OPERAND

Issued by: \$EDXASM

Explanation: The system encountered invalid numeric data while attempting to convert an EBCDIC character string to a binary value using the CONVTD instruction.

System Action: The error is flagged and compilation continues.

User Response: Examine the EBCDIC field for invalid characters and make the appropriate corrections.

**INVALID NUMERIC RANGE ON COMMON=
OPERAND, MUST BE 0 TO 31**

Issued by: \$EDXASM

Explanation: During system generation, you coded an invalid parameter for the COMMON operand on the SYSTEM statement. When specifying the size (in 2K blocks) of the common area in a partition, you must code a value from 0 to 31.

System Action: The error is flagged and compilation continues.

User Response: Ensure that all values coded for the COMMON operand are within the valid range.

**INVALID OPERAND 3. MUST BE 'ON' OR
'OFF'**

Issued by: \$EDXASM

Explanation: For the SETBIT instruction, the third operand must be 'on' or 'off'.

System Action: Continues processing but does not generate object text for that instruction.

User Response: Correct the indicated error and reassemble.

INVALID OPERAND 3. MUST BE 'SECS'

Issued by: \$EDXASM

Explanation: For the STIMER instruction, the third operand must be SECS.

System Action: The error is flagged and compilation continues.

User Response: Correct the indicated error and recompile the program.

INVALID OPERAND ON EXEC COMMAND

Issued by: \$JOBUTIL

Explanation: In the procedure data set, you have entered invalid information.

System Action: Ends utility.

User Response: Edit the procedure data set, taking out the invalid information and placing all elements in the proper order. Then retry the utility.

INVALID OPERAND SPECIFIED ON 'TERMCTRL'

Issued by: \$EDXASM

Explanation: An invalid operand or combination of operands was coded in the TERMCTRL statement.

System Action: The error is flagged and compilation continues.

User Response: Examine the TERMCTRL statement and make the needed corrections.

INVALID OPERATION FOR VIRTUAL TERMINAL

Issued by: \$TERMUT1

Explanation: You tried to vary off a virtual terminal. This is invalid.

System Action: Returns to command mode.

User Response: This is an informational message. No user response is needed.

INVALID OPTION

Issued by: \$IAMUT1

Explanation: Using the AL subcommand of DR, the directory data set already exists, and your response to the allocation option was other than AS, BN, EN or a null.

System Action: Displays the valid options and asks you to enter one.

User Response: Reply with a valid response.

INVALID OPTION ENTERED

Issued by: \$IAMUT1

Explanation: You entered an option other than 0 through 3.

System Action: Asks you to enter another option.

User Response: Enter either 0, 1, 2, or 3 as an option.

INVALID OR MISSING COUNT= SPECIFICATION

Issued by: \$EDXASM

Explanation: The value specified for the number of queue elements to be generated was coded incorrectly or omitted. An equated value cannot be used for count.

System Action: The error is flagged and compilation continues.

User Response: Specify an explicit positive integer for the COUNT operand on the DEFINEQ instruction.

INVALID OR MISSING DEVICE TYPE SPECIFIED

Issued by: \$EDXASM

Explanation: A device was not specified or was specified incorrectly during system configuration.

System Action: The error is flagged and compilation continues.

User Response: Specify the device correctly on the TERMINAL statement or include the missing device.

INVALID OR MISSING DI, DO, OR PI ADDRESS - ALL MUST BE PRESENT

Issued by: \$EDXASM

Explanation: For the 4013 terminal, the DI, DO, and PI operands must be specified.

System Action: The error is flagged and compilation continues.

User Response: Specify all values and in the form:

DI=(address,termaddr)
DO=(address,termaddr)
PI=(address,bit)

Refer to the *Installation and System Generation Guide* for details on how to specify the TERMINAL statement for the 4013.

INVALID OR MISSING LABEL ON 'ADAPTER' STATEMENT

Issued by: \$EDXASM

Explanation: All ADAPTER statements must have labels. A valid label cannot exceed 8 characters.

System Action: The error is flagged and compilation continues.

User Response: Specify a 1–8 character name as a label for each ADAPTER statement.

INVALID OR MISSING SIZE= SPECIFICATION

Issued by: \$EDXASM

Explanation: The value specified for the buffer size was coded incorrectly or omitted.

System Action: The error is flagged and compilation continues.

User Response: Specify the buffer size as a self-defining term for the SIZE operand on the DEFINEQ instruction.

INVALID OR MISSING TYPE= PARAMETER

Issued by: \$EDXASM

Explanation: The parameter for TYPE= on the ADAPTER statement was missing or specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify TYPE=MFA or TYPE=ALPA on the ADAPTER statement.

INVALID OR UNDEFINED OPERATION CODE

Issued by: \$EDXASM

Explanation: The instruction was not found.

System Action: The error is flagged and compilation continues.

User Response: Ensure the instruction you specified is spelled correctly or that no blank lines appear in the source code.

INVALID OR UNSPECIFIED LENGTH OPERAND

Issued by: \$EDXASM

Explanation: The LENGTH operand is required in an SLE instruction and must be a self-defining term.

System Action: The error is flagged and compilation continues.

User Response: Code the LENGTH operand if it has not been coded or recode the existing operand with a self-defining term.

INVALID PARAMETER COUNT

Issued by: \$EDXASM

Explanation: Too few or too many operands were specified in an instruction.

System Action: The error is flagged and compilation continues.

User Response: On the instruction in which the error occurred, code a valid number of operands.

INVALID PARAMETER IN DATA LIST

Issued by: \$EDXASM

Explanation: The contents of the list operand in a GETEDIT or PUTEDIT instruction were incorrectly specified.

System Action: The error is flagged and compilation continues.

User Response: Check the list operand in the appropriate instruction and correct the format.

INVALID PARAMETER PASSED - DEFAULT TO NO ERROR MSG DISPLAY ON OCCURRENCE

Issued by: \$LOG

Explanation: The program or \$JOBUTIL procedure that loaded the \$LOG utility passed an invalid parameter. \$LOG ignores the parameter and uses the default. If the utility encounters an error, such as the log data set being full, it will not display a message. You must use the \$LOG command to display the utility's error messages.

System Action: The program or \$JOBUTIL procedure continues to execute.

User Response: If you want \$LOG to display utility error messages when they occur, code the correct parameter and reload the program or \$JOBUTIL procedure.

INVALID PARAMETER PASSED - DEFAULT TO NO TERMINATE ON DATA SET WRAP

Issued by: \$LOG

Explanation: The program or \$JOBUTIL procedure that loaded the \$LOG utility passed an invalid parameter. \$LOG ignores the parameter and uses the default. If the log data set becomes full during I/O error logging, \$LOG will return to the third record in the data set and begin writing over the existing entries.

System Action: The program or \$JOBUTIL procedure continues to execute.

User Response: If you want \$LOG to end before it writes over any existing entries in the log data set, code the correct parameter and reload the program or \$JOBUTIL procedure.

INVALID PARAMETER WITH 'ALIGN' OPERAND

Issued by: \$EDXASM

Explanation: The "type" operand specified on the ALIGN instruction is not one of the valid data alignment types.

System Action: The error is flagged and compilation continues.

User Response: Specify for "type" either WORD, BYTE, or DWORD. If omitted, WORD is the default.

INVALID PARM= SPECIFICATION

Issued by: \$EDXASM

Explanation: The operand PARM= on the PROGRAM statement was either too large or was not specified as a positive self-defining word count.

System Action: The error is flagged and compilation continues.

User Response: Ensure the value specified for PARM= is a positive self-defining word count. The number of words specified in the operand list must not exceed 762 words less 33 for each data set (DS=) and each overlay (PGMS=).

INVALID PGM= SPECIFICATION

Issued by: \$EDXASM

Explanation: An error in the PGM= operand was detected while trying to construct the DSCB data for the PROGRAM instruction.

System Action: The error is flagged and compilation continues.

User Response: Ensure the PGM= operand is coded correctly.

INVALID PRECISION FOR IMMEDIATE OPERAND 2

Issued by: \$EDXASM

Explanation: The immediate operand 2 cannot have byte specification.

System Action: The error is flagged and compilation continues.

User Response: Operand 2 must be single precision if specified as immediate data.

INVALID 'POINT=', MUST BE '0 - 15' FOR AI OR '0 - 1' FOR AO

Issued by: \$EDXASM

Explanation: The value specified for the POINT= operand on the IODEF instruction was omitted or specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify a value 0–15 for analog input (AI) or 0–1 for analog output (AO).

INVALID PREC= SPECIFICATION

Issued by: \$EDXASM

Explanation: The precision coded for the PREC= operand on the READ/WRITE was specified incorrectly. Must be either PREC=S or PREC=D.

System Action: The error is flagged and compilation continues.

User Response: Specify either "S" or "D" for PREC=. Also, if PREC=D is specified, the "relrecno" (relative record number) operand must be coded.

INVALID PRECISION FOR REGISTER OPERATION

Issued by: \$EDXASM

Explanation: Only single precision arithmetic operations can occur with direct register specification. Addressing through registers is valid. For example, #reg cannot be specified for other than single precision (word) arithmetic operations. (n,#reg) can be used for all precisions.

System Action: The error is flagged and compilation continues.

User Response: Correct the error and recompile.

INVALID PRIORITY SPECIFICATION

Issued by: \$EDXASM

Explanation: The priority assigned to the task in a TASK statement was not a self-defining term or was outside the accepted range of 1 (highest priority) to 510 (lowest priority).

System Action: The error is flagged and compilation continues.

User Response: Ensure that the priority coded in the TASK statement is a self-defining term within the accepted range.

INVALID PROCEDURE NAME xxx

Issued by: \$JOBUTIL

Explanation: This problem can occur if you have loaded the utility with the \$L command, or through the Session Manager. One or more of the names you specified in the PROC statement were wrong. Either the data set does not exist on the volume specified, or the volume does not exist, or both. It is also possible that you did not enter the information starting in column 10.

System Action: Ends the operation.

User Response: Retry the function and specify valid information.

INVALID PROGRAM NAME

Issued by: \$PREFIND

Explanation: The data set you specified to contain the program does not exist.

System Action: Returns to command mode.

User Response: Specify an existing program-type data set, or allocate one.

INVALID PROGRAM NAME

Issued by: \$MTM

Explanation: The name of the program requested from the primary menu was not found in the Multiple Terminal Manager program table or invalid parameters were supplied on a DISCONNECT command.

System Action: Does not perform the requested function.

User Response: Correct the program name or parameters and retry the request.

INVALID PROMPT MESSAGE

Issued by: \$EDXASM

Explanation: The prompting message on the GETVALUE instruction was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the label of the TEXT statement containing the prompting message or specify an explicit character string enclosed in quotes.

INVALID PROMPT= SPECIFICATION

Issued by: \$EDXASM

Explanation: The PROMPT= operand for GETVALUE and READTEXT instructions was specified incorrectly. Valid operands are PROMPT=COND (advance input accepted), and PROMPT=UNCOND (prompt is displayed).

System Action: The error is flagged and compilation continues.

User Response: Specify either COND or UNCOND with PROMPT=. If omitted, PROMPT=UNCOND is the default.

INVALID 'PULSE' SPECIFIED

Issued by: \$EDXASM

Explanation: The form of the PULSE operand for digital output on the SBIO instruction was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the PULSE operand as follows:

SBIO DOx, (PULSE, dir)

where "dir" (direction) is either ON (or UP) or OFF (or DOWN).

**INVALID 'RANGE=', MUST BE 5V, 500MV,
200MV, 100MV, 50MV, 20MV, OR 10MV**

Issued by: \$EDXASM

Explanation: The value you specified for the RANGE= operand on the IODEF instruction is incorrect.

System Action: The error is flagged and compilation continues.

User Response: Specify either 5V, 500MV, 200MV, 100MV, 50MV, 20MV, or 10MV as the value for RANGE=.

**INVALID RELATIONAL OPERATOR
SPECIFIED ON STATEMENT SYNTAX**

Issued by: \$EDXASM

Explanation: An IF or DO statement was formatted incorrectly, or a relational operator within the structure was invalid.

System Action: The error is flagged and compilation continues.

User Response: Examine the statement syntax and the relational operator(s) used and make the appropriate corrections.

INVALID RESET TYPE

Issued by: \$RJE2780, \$RJE3780

Explanation: The reset type you entered is not one of the valid types: CO, SU, SP, or PU.

System Action: Utility returns to wait state (it is invoked with ATTNLIST processing).

User Response: Retry the function, entering a valid command.

INVALID SECOND CHARACTER IN PREC=

Issued by: \$EDXASM

Explanation: The second character in the PREC= specification must be either F (single precision), L (extended precision), or * (single precision as the default) for floating-point arithmetic operations.

System Action: The error is flagged and compilation continues.

User Response: Specify the second character for PREC= as explained.

INVALID SELF-DEFINING TERM

Issued by: \$EDXASM

Explanation: \$EDXASM expected to find a self-defining term but has not found it. This error can occur if an instruction has an incorrect syntax and \$EDXASM is reading the wrong part of a character string.

System Action: The error is flagged and compilation continues.

User Response: Examine the syntax of the instruction and make any needed corrections. This error message also can be triggered by previous errors in the program.

INVALID SENSOR INPUT/OUTPUT TYPE

Issued by: \$EDXASM

Explanation: The sensor input/output type for the SBIO instruction was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify the input/output type as 3 or 4 characters in the format: AIx, AOx, DIx, DOx — where x is a one- or two-digit number.

INVALID 'SEQ=' FOR 'AI'

Issued by: \$EDXASM

Explanation: The parameter coded for the SEQ= operand was specified incorrectly on the SBIO instruction. For type AI, the parameter for the SEQ= operand must be either YES or NO.

System Action: The error is flagged and compilation continues.

User Response: Specify either SEQ=YES or SEQ=NO as appropriate.

INVALID SESSION PARAMETERS RECEIVED - BIND REJECTED

Issued by: \$RJESNA

Explanation: The session parameters (BIND image) received from the host are not valid for the Event Driven Executive SNA support.

System Action: The SNA support rejects the received session parameters and \$RJESNA terminates.

User Response: =x \$RJESNA requires native and host support. You may have to consult the programmer who wrote the host interface to help solve this problem. After the problem is corrected, restart \$RJESNA.

INVALID SIGNON CHARACTER

Issued by: \$MTM

Explanation: The SIGNON specification for the TERMINAL file record listed immediately before this message is not "Y" or "N".

System Action: The terminal is not connected.

User Response: Edit the TERMINAL record to correct the SIGNON specification. Stop and restart the manager.

INVALID SIZE FOR AL

Issued by: \$JOBUTIL

Explanation: In the AL command, you specified an invalid size.

System Action: Ends utility.

User Response: Edit the procedure data set, putting the elements in the proper order. Then rerun the utility.

INVALID SPECIFICATION FOR ECHO

Issued by: \$EDXASM

Explanation: The valid parameters for the ECHO= operand are "YES" and "NO". Refer to the TERMINAL statement in the *Installation and System Generation Guide* to determine which parameters are valid for this device.

System Action: The error is flagged and compilation continues.

User Response: Specify either "YES" or "NO" for the ECHO= operand.

INVALID SPECIFICATION FOR 'OVFLINE'

Issued by: \$EDXASM

Explanation: The parameter specified for the OVFLINE= operand (line overflow) on the TERMINAL statement must be either "YES" or "NO".

System Action: The error is flagged and compilation continues.

User Response: Specify either OVFLINE=YES or OVFLINE=NO.

INVALID SPECIFICATION FOR 'SCREEN'

Issued by: \$EDXASM

Explanation: The parameter specified for the SCREEN= operand on the TERMINAL statement was not "YES", "ROLL", or "NO" for 4013, 2741, TTY, or GPIB devices. For 4978/4979 or ACCA devices, the parameter specified was not "YES", "ROLL", "NO", or "STATIC".

System Action: The error is flagged and compilation continues.

User Response: Specify the SCREEN= operand according to the device type.

INVALID SYNTAX IN OPERAND FIELD

Issued by: \$EDXASM

Explanation: The syntax of one of the operands is invalid for this instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the correct syntax for the operands of this instruction.

INVALID TASK NAME

Issued by: \$DEBUG

Explanation: You specified the name of a task that does not exist in the program you are debugging.

System Action: Awaits operator response.

User Response: Reenter the information using the name of a valid task for the program you are debugging.

INVALID TCB LABEL ON 'TCBGET' OR 'TCBPUT'

Issued by: \$EDXASM

Explanation: Operand 2 in a TCBGET or TCBPUT instruction is not a valid pointer to a TCB field.

System Action: The error is flagged and compilation continues.

User Response: Code a valid TCB label.

INVALID TERMINAL

Issued by: \$MTM

Explanation: The terminal name entered with a DISCONNECT command is not a Multiple Terminal Manager terminal.

System Action: Does not disconnect the terminal.

User Response: Retry, specifying a valid terminal name.

INVALID 'TERMINAL NAME', MUST BE 1 - 8 CHARACTERS

Issued by: \$EDXASM

Explanation: The terminal name specified exceeds the maximum of eight characters.

System Action: The error is flagged and compilation continues.

User Response: Ensure the terminal name does not exceed eight characters.

INVALID THIRD CHARACTER IN PREC=

Issued by: \$EDXASM

Explanation: The third character in the PREC= specification must be either F (single precision), L (extended precision), or * (single precision as the default) for floating-point arithmetic operations.

System Action: The error is flagged and compilation continues.

User Response: Specify the third character for PREC= as explained.

INVALID TIMEOUT OR CHAIN OPERAND

Issued by: \$EDXASM

Explanation: If the TIMEOUT and CHAIN operands of the BSCREAD instruction are specified, the only valid parameters are either "YES" or "NO".

System Action: The error is flagged and compilation continues.

User Response: Specify the valid parameter as appropriate.

INVALID TOP MARGIN

Issued by: \$EDXASM

Explanation: The syntax for the TOPM= operand of the TERMINAL statement was specified incorrectly. The value must be a positive self-defining term. The value must be between zero and PAGESIZE-1.

System Action: The error is flagged and compilation continues.

User Response: Specify a positive self-defining term for the TOPM= operand.

INVALID TYPE FOR AL

Issued by: \$JOBUTIL

Explanation: You specified an invalid data set type in the AL command.

System Action: Ends utility.

User Response: Edit the procedure data set, putting the elements in the proper order. Then retry the utility.

INVALID TYPE INDICATOR - xxx ON LINE #yyy

Issued by: \$MSGUT1

Explanation: Invalid type specified in the operand.

System Action: Continues processing.

User Response: When processing finishes, correct the error and retry.

INVALID 'TYPE=', MUST BE 'DATA' OR 'ALL'

Issued by: \$EDXASM

Explanation: An invalid parameter was specified for the TYPE= operand of the ERASE instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify either "DATA" or "ALL" as the operand for TYPE=.

INVALID "TYPE" OPERAND

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the TYPE operand of the NETCTL instruction.

System Action: The error is flagged and compilation continues.

User Response: Code one of the following parameters for the TYPE operand:

- RECV
- ACCEPT
- REJECT
- CANCEL
- QEC
- RELQ
- SIG
- LUSTAT
- RTR

INVALID 'TYPE=' SPECIFIED

Issued by: \$EDXASM

Explanation: The operand specified for TYPE= is invalid or its parameter is specified incorrectly on the TERMCTRL instruction for GETSTORE or PUTSTORE functions.

System Action: The error is flagged and compilation continues.

User Response: Refer to the description of the TERMCTRL instruction for the parameters that can be coded for a specific device.

INVALID VALUE

Issued by: \$TERMUT2

Explanation: The number you supplied in response to a prompt is not in the valid range or format.

System Action: Prompts for a valid number.

User Response: Reenter a valid hex number for the function being run.

INVALID VOLUME LABEL

Issued by: \$TAPEUT1

Explanation: The volume label you specified is in an unacceptable format for use with the allocate function.

System Action: Awaits operator response.

User Response: Specify a volume label with an acceptable format.

INVALID VOLUME LABEL

Issued by: \$XPSLINK from \$XPSPOST

Explanation: In response to a prompt for an output program data set, the volume name you entered was too long (more than 6 characters).

System Action: Asks again for the output program data set name and volume.

User Response: Specify a volume name with 1 - 6 characters.

INVALID VOLUME LABEL

Issued by: \$PREFIND

Explanation: The volume you specified does not exist.

System Action: Returns to command mode.

User Response: Specify a correct volume name (6 letters, starting with a letter) and retry the function.

INVALID VOLUME LABEL ON *COPYCOD LABEL IN \$EDXL

Issued by: \$EDXASM

Explanation: The volume name specified on the *COPYCOD statement was not found.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the volume name specified on the *COPYCOD statement is correct.

INVALID WRES SUBCOMMAND

Issued by: Spooling

Explanation: An invalid subcommand has been specified on a \$\$ WRES command.

System Action: Does not perform the requested function.

User Response: Retry the function with valid subcommands.

INVALID WRITER NAME

Issued by: Spooling

Explanation: A device not designated as a spool device has been specified on a \$\$ WSTR, WRES, or WSTP command, or the writer is already active.

System Action: Does not perform the requested function.

User Response: Retry the command with a correct device specified. Use the \$\$ DISP ALL command to determine the valid device names.

INVALID WSTP SUBCOMMAND

Issued by: Spooling

Explanation: An invalid subcommand has been specified on a \$\$ WSTP command.

System Action: The requested function is not performed.

User Response: Retry the function with valid subcommands.

INVALID 'ZCOR=', MUST BE 'YES' OR 'NO'

Issued by: \$EDXASM

Explanation: The parameter specified for the ZCOR= operand (zero correction facility) on the IODEF instruction was coded incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify either "YES" or "NO" for ZCOR= and code it only for analog input (AI).

INVALID 4982 BASE ADDRESS

Issued by: \$EDXASM

Explanation: The base address for a 4982 sensor I/O device must end in 0 or 8.

System Action: The error is flagged and compilation continues.

User Response: Recode the base address in the ADDRESS operand of the SENSORIO statement.

I/O ERROR ACCESSING DATA SET

Issued by: \$TERMUT2

Explanation: A read or write error occurred in the output data set.

System Action: Returns to command mode.

User Response: Recreate the input data set and retry the function.

INVALID 4982 FEATURE ADDRESS

Issued by: \$EDXASM

Explanation: The addresses specified for one or more of the operands, AI, AO, DI, DO and PI, are not within the range of the base address for the 4982 device.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the addresses specified for the operands fall within the range of the base address for the 4982 device.

I/O BUFFER ADDRESS NOT SPECIFIED

Issued by: \$EDXASM

Explanation: The "loc" operand on the READTEXT instruction must be coded. It specifies the label of the buffer storage area that is to receive data upon execution of the READTEXT.

System Action: The error is flagged and compilation continues.

User Response: Specify the label of the storage area to receive the data.

I/O ERROR ATTEMPTING TO OPEN SEGMENT OR IPL VOL

Issued by: \$XPSLINK from \$XPSPOST

Explanation: The utility could not open the volumes needed to perform the requested function.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Rerun \$XPSLINK. If the problem occurs again, there is probably a bad spot on the disk or diskette in the IPL or work volume in use. Use \$INITDSK to verify those two volumes.

I/O ERROR DURING SEARCH FOR VOLUME xxx

Issued by: \$INITDSK

Explanation: A read/write error occurred while the utility was looking for the specified volume.

System Action: Returns an error code. Ends utility.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Then retry the function. You may have to wait until the system is not in use by anyone else. If the problem persists, you may have to reinitialize the volume or device.

I/O ERROR DURING DSOPEN - RC = xxx

Issued by: \$TAPEUT1

Explanation: An error prevented the opening of a tape file.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

I/O ERROR DURING OPEN

Issued by: \$TAPEUT1

Explanation: An I/O error prevented the opening of the data set you specified.

System Action: Issues a tape read/write return code. Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**I/O ERROR INITIALIZATION FIXED HEAD
DEV, DISK RETURN CODE=xxx IN THE
TWO RECORDS STARTING WITH
RECORD yyy**

Issued by: \$DASDI

Explanation: An error was encountered during fixed head initialization starting with record yyy. The result was a return code of xxx.

System Action: Terminates the initialization.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**I/O ERROR DURING OPEN OF COPYCODE
DATA SET**

Issued by: \$EDXASM

Explanation: An I/O error occurred during the directory lookup for the copy code data set specified or the data set was not found in the directory.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the copy code data set name is specified correctly and that it is present in at least one of the copy code volumes.

**I/O ERROR ON S/1-S/1 - STATUS =xxx
HEX yyy DEC**

Issued by: \$S1S1UT1

Explanation: A read/write error occurred on the Series/1. The result was a return code of yyy (decimal value).

System Action: Returns to command mode.

User Response: Check the Series/1-to-Series/1 return code to find the cause of the problem, and take appropriate action. Then retry the function.

I/O ERROR RC = xxx ON VOLUME

Issued by: \$MOVEVOL

Explanation: A read or write error occurred on the diskette in use.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Retry the function using another diskette.

I/O ERROR RC =xxx

Issued by: \$TAPEUT1

Explanation: An error occurred during the opening of the specified source or target data sets.

System Action: Prompts, "REDEFINE? (Y,N):".

User Response: Answer the prompt with 'Y' and specify different data areas.

I/O ERROR RC =xxx CONTENTS OF BUFF:

Issued by: \$TAPEUT1

Explanation: A read or write error occurred while exercising the tape.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**I/O ERROR READING \$SMMAIN \$JOBUTIL
FILE xxx RC= yyy**

Issued by: Session Manager

Explanation: While trying to submit a background job, the session manager encountered an error while reading file \$SMWxxx on volume EDX003 (xxx is your user ID).

System Action: The session manager does not submit the job. It returns to the previous menu.

User Response: Check the disk and diskette read/write return codes for the cause of the error. You also can end the session manager and respond "N" to the question "Do you want to save work data sets (Y/N)?".

'IOTYPE=' MUST BE 'INPUT' OR 'OUTPUT'

Issued by: \$EDXASM

Explanation: The only valid parameters for the IOTYPE= operand are "INPUT" and "OUTPUT".

System Action: The error is flagged and compilation continues.

User Response: Specify either "INPUT" or "OUTPUT" as the parameter for IOTYPE= on the DCB statement. If omitted, IOTYPE=OUTPUT is the default.

**I/O ERROR WRITING \$JOBQ \$JOBUTIL
FILE RC= xxx**

Issued by: Session Manager

Explanation: While trying to submit a background job, the session manager encountered an error while writing to a temporary \$JOBQ file on EDX003.

System Action: The session manager does not submit the job. It returns to the previous menu.

User Response: Check the disk and diskette read/write return codes for the cause of the error and take the appropriate action.

IPL NUCLEUS HAS AN INVALID HEADER

Issued by: \$S1S1UT1

Explanation: The utility has found invalid data in the target nucleus.

System Action: Returns to command mode.

User Response: Resolve the problem with the nucleus in question. Then retry.

IPL TEXT I/O ERROR, DISK RETURN CODE = xxx

Issued by: \$INITDSK

Explanation: A read/ write error occurred in the IPL text.

System Action: Returns to command mode.

User Response: End the utility, reload and retry the function. If the problem occurs again, re-IPL and retry.

IPL VOLUME CANNOT BE FIXED HEAD VOLUME

Issued by: \$INITDSK

Explanation: You specified a fixed-head volume for use as the IPL volume. An IPL volume can only be a non-fixed-head volume.

System Action: Ends the operation.

User Response: Specify a non-fixed-head volume as the IPL volume.

IS THIS A SECONDARY ENTRY (Y/N)?

Issued by: \$IAMUT1

Explanation: This is a prompt (using the IE subcommand of DR) for information about the entry being inserted into the directory.

System Action: Awaits operator response.

User Response: Reply "Y" if you are inserting a secondary entry, so you can be prompted for the data set name and volume of the primary indexed file. Reply "N" if you are inserting a primary entry.

JOB xxx NOT FOUND ON JOB QUEUE DATA SYSTEM

Issued by: \$SUBMIT

Explanation: The job number you specified does not exist.

System Action: Returns to command mode.

User Response: Find the correct job number by using the STQ command. Then retry using a valid job number.

JOB xxx WAS EXECUTING WHEN A SYSTEM FAILURE OCCURRED IT HAS BEEN PUT IN A HELD STATE

Issued by: \$JOBQUT

Explanation: The held job may have caused the system to fail. It will start executing again when the system comes back up.

System Action: None.

User Response: Before restarting or releasing the job, determine that it did not cause the system to fail.

JOBID INVALID

Issued by: Spooling

Explanation: The specified spool job is not in the system.

System Action: Does not perform the requested function.

User Response: Determine the correct spool job ID and resubmit the command. The \$\$ DISP ALL command can be used to display the current spool jobs.

JOB PRIORITY INVALID

Issued by: \$SUBMIT

Explanation: You attempted to assign a job priority that is outside the valid range or that is in the wrong format.

System Action: Issues message stating valid range and prompts for a correct job priority.

User Response: Specify a job priority in the range of 0 - 3.

JOB QUEUE PROCESSOR LOAD FAILED RC = xxx

Issued by: \$SUBMIT, \$JOBQUT

Explanation: The system cannot load the job queue processor at this time.

System Action: Ends the operation.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action. Then retry the function.

JOURNAL ABORTED - FILE ERROR xxx

Issued by: \$RJESNA

Explanation: An error occurred while the JOURNAL function attempted to open or write to the Series/1 data set. The result was a return code of xxx.

System Action: Terminates the JOURNAL request.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

JOURNAL ON/OFF REQUESTED - ROUTINE NOT AVAILABLE

Issued by: \$RJESNA

Explanation: You entered a JOURNAL command, but the journal processing routines were not included when \$RJESNA was link-edited.

System Action: None.

User Response: To perform journal processing, link-edit \$RJESNA again and include the journal processing routines distributed with the \$RJESNA utility. After the link-edit, restart the \$RJESNA utility and then enter the JOURNAL commands.

JOURNAL TERMINATED

Issued by: \$RJESNA

Explanation: This message is issued for one of the following reasons:

1. You entered a JOURNAL OFF command.
2. The JOURNAL facility failed to find the data set or volume you specified. A previous message will have identified the problem.
3. An error occurred in opening the specified data set. A previous message will have identified the error condition.
4. An I/O error occurred on the journal data set. A previous message will have identified the error condition.

System Action: Terminates JOURNAL processing.

User Response: If \$RJESNA has issued a previous error message, examine its explanation and take the appropriate corrective action.

KEY OF INPUT REC xxx IS DUPLICATE OR OUT OF SEQUENCE. OMIT THE RECORD AND CONTINUE?

Issued by: \$IAMUT1

Explanation: A duplicate key exists in the input sequential data set xxx and could not be written to indexed data set.

System Action: Waits for response.

User Response: If "Y" is entered, the next record is read from input data set and processing continues. If "N" is entered, the command terminates. Check that the input data set contains the proper data. Check the KEYSIZE and KEYPOS operands used to define the indexed data set against the input data set records. Check that the data in the input data set is in the proper sequence. Redefine and reload data set if necessary.

KEYWORD SSCPID HAS TO BE 0 - 12 HEX CHARACTERS

Issued by: \$EDXASM

Explanation: You did not specify a value of 0–12 hexadecimal characters for the SSCPID operand of the NETHOST instruction.

System Action: The error is flagged and compilation continues.

User Response: Recode the SSCPID operand with a valid number of hexadecimal characters.

LABEL IS TOO LONG OR IS MISSING

Issued by: \$EDXASM

Explanation: The label is required if ADAPTER=MFA or ADAPTER=ALPA; otherwise the label is optional. Valid labels cannot exceed 8 characters.

System Action: The error is flagged and compilation continues.

User Response: Specify 1–8 characters for the label.

LABEL= MUST BE EITHER SL, NL, OR BLP (DEFAULT=SL)

Issued by: \$EDXASM

Explanation: The parameter specified for the LABEL= operand of the TAPE statement must be either “SL” (standard label), “NL” (no label), or “BLP” (bypass label processing).

System Action: The error is flagged and compilation continues.

User Response: Specify either “SL”, “NL”, or “BLP” for the LABEL= operand. If omitted, LABEL=SL is the default.

LABEL MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The label field required for this instruction was not coded.

System Action: The error is flagged and compilation continues.

User Response: Specify a 1–8 character name as the label on this instruction.

LABEL MUST START WITH AN ALPHABETIC OR SPECIAL CHARACTER

Issued by: \$EDXASM

Explanation: Valid labels must have as the first character: A—Z, \$, #, or @.

System Action: The error is flagged and compilation continues.

User Response: Specify any of the valid characters as the first character of the label.

LABEL NOT SUPPLIED - CAIOCB TERMINATED

Issued by: \$EDXASM

Explanation: A label on the CAIOCB statement was not supplied.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a label, and reassemble.

LENGTH IGNORED IN TEXT FIELD ONLINE

x. x

Issued by: \$MSGUT1**Explanation:** Length should not be used in text (T) type parameter.**System Action:** Continues processing.**User Response:** When processing finishes, correct the error and retry.**LENGTH NOT SPECIFIED IN CHARACTER FIELD ON LINE #xxx****Issued by:** \$MSGUT1**Explanation:** The character type parameter must specify length.**System Action:** Continues processing.**User Response:** When processing finishes, correct the error and rerun the function.**'LEVEL' MUST BE EITHER 0, 1, 2, OR 3****Issued by:** \$EDXASM**Explanation:** The hardware interrupt level specified to be assigned to a device by a PREPARE instruction is invalid on the IDCBC instruction.**System Action:** The error is flagged and compilation continues.**User Response:** Specify 0, 1, 2, or 3 for the hardware interrupt level on the LEVEL= operand.**LIBRARY IS FULL****Issued by:** \$DISKUT1**Explanation:** No data set directory space is left for the volume in use.**System Action:** Returns to command mode.**User Response:** Delete some unneeded data sets or use a different volume.**LINE LEVEL FOR THIS DEVICE:****Issued by:** \$GPIBUT1**Explanation:** The system is asking for a line level value of either 1 or 0.**System Action:** Awaits operator response.**User Response:** Enter the desired value.**LINE MODE TYPE NOT SUPPORTED****Issued by:** \$EDXASM**Explanation:** The line mode type must be one of the following: RS422, LOCAL, PTTOPT, or SWITCHED.**System Action:** The error is flagged and compilation continues.**User Response:** Specify the correct line mode type in the LMODE operand of the TERMINAL statement.

LINK LEVEL ACTIVATED VIA SNA**Issued by:** \$SNAMAIN**Explanation:** Series/1 SNA support has been loaded and the resources required to operate the network have been obtained. The primary SDLC station has set the Series/1 SDLC station to normal response mode.**System Action:** Normal processing continues.**User Response:** This is an informational message. No user response is needed.**'LISTADDR' POSITIONAL PARAMETER IS MISSING****Issued by:** \$EDXASM**Explanation:** The "listaddr" operand of the EXOPEN instruction is required and must specify the address (the label) of the first word of a list of three addresses. Briefly the addresses are:

- word1 — address of storage area for interrupt processing
- word2 — address of a list of ECB addresses
- word3 — DCB address

Refer to the *Language Reference* for a complete description of the operand.**System Action:** The error is flagged and compilation continues.**User Response:** Specify the "listaddr" operand as explained.**LMODE=LOCAL/PTTOPT REQUIRED BY 4975-R****Issued by:** \$EDXASM**Explanation:** The line mode (LMODE=) specification for the 4975-01R and 4975-02R device must be either "LOCAL" or "PTTOPT".**System Action:** The error is flagged and compilation continues.**User Response:** Specify either "LOCAL" or "PTTOPT" for the LMODE= operand of TERMINAL statement.**'LMODE=LOCAL/PTTOPT/SWITCHED' MUST BE ON ADAPTER BASE ADDRESS****Issued by:** \$EDXASM**Explanation:** If ADAPTER=MFA is specified, and LMODE= LOCAL, PTTOPT, or SWITCHED, the address specified on the ACCA TERMINAL statement must be an MFA (multifunction adapter) base address (the address is divisible by 4). If LMODE=RS422, the terminal can be connected to any port of the MFA.**System Action:** The error is flagged and compilation continues.**User Response:** Specify an MFA-based address on the TERMINAL statement.

LOAD ERROR ENCOUNTERED, RX = xxx

Issued by: \$XPSLINK

Explanation: \$XPSLINK was unable to load \$XPSPRE, \$EDXLINK, or \$XPSPOST. The result was a LOAD return code of xxx.

System Action: \$XPSLINK ends.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action.

LOADER ERROR RC = xxx

Issued by: \$DICOMP

Explanation: A load error occurred during the requested function, resulting in a return code of xxx.

System Action: Ends the operation.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action. Retry the function.

**LOADER ERROR WHILE PROCESSING
TERMINAL STATEMENT**

Issued by: \$EDXASM

Explanation: There is no room in the partition for a second level overlay which the TERMINAL statement is attempting to read. Second level overlays process individual TERMINAL devices.

System Action: The error is flagged and compilation continues.

User Response: Recompile the program after ensuring that there is enough room in the partition.

LOAD ERROR RC=xxx

Issued by: \$MTM

Explanation: A load failure occurred, resulting in a LOAD return code of xxx.

System Action: The terminal is not available to the Multiple Terminal Manager.

User Response: Check the LOAD return code to find the cause of the problem and take the appropriate action.

**LOAD ERROR RC = xxx JOB PROCESSOR
ENDED**

Issued by: \$JOBUTIL

Explanation: The job queue processor was unable to load the program you requested. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action. Then try to load the program again.

LOAD FAILED (\$DISKUT3), RC = xxx

Issued by: \$IMAGE

Explanation: The requested function required the system to load \$DISKUT3, which it could not do. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the \$DISKUT3 return code to find the cause of the problem, and take appropriate action. Retry the function.

**LOAD FAILED FOR \$IAMSTGM, RC=xxx,
PTN=yyy, SIZE=zzz**

Issued by: \$IAM

Explanation: \$IAM attempted to load \$IAMSTGM in partition yyy, whose size is zzz. The result was a LOAD return code of xxx.

System Action: Data paging initialization will continue. If you requested storage in other partition(s) and some or all of that storage is obtained and the total obtained is at least 4k, data paging will be activated.

User Response: To correct the size of the page area, cancel \$IAM and \$IAMSTGM, make sure the storage is available in the partition you requested and reload \$IAM.

LOAD FOR SERVER xxx FAILED, RC=yyy

Issued by: \$MTM

Explanation: A load failure occurred during initialization for the server xxx, resulting in a LOAD return code of yyy.

System Action: The terminal is not available to the Multiple Terminal Manager

User Response: Check the LOAD return code to find the cause of the problem and take the appropriate action.

LOAD OF \$SNAMAIN FAILED — RC= xxx

Issued by: \$SNA

Explanation: During network activation, \$SNA failed to load \$SNAMAIN.

System Action: \$SNA unloads and issues the return code xxx.

User Response: Check the LOAD instruction return codes for an explanation of the code received in this message and take the appropriate action. If the return code is 76 or 80, ignore the return code explanation and refer to the installation procedures in the SNA Program Install Directory.

**LOG DATA SET WRAPPED xxx TIMES,
LAST RECORD LOGGED BEFORE WRAP
= yyy**

Issued by: \$LOG

Explanation: The log data set has been “wrapped” xxx times. “Wrapping” occurs when the data set becomes full during I/O error logging. \$LOG returns to the third record in the log data set and begins writing over the existing entries. The last record logged before the utility wrapped the data set is yyy. The value yyy is also the number of usable log records in the log data set. This value does not include the first two records of the data set which contain control information.

System Action: I/O error logging continues.

User Response: Use the \$DISKUT2 utility to display or print the existing error log records.

LOGGING NOT INCLUDED IN SUPERVISOR

Issued by: \$LOG

Explanation: You attempted to use the logging function, but your system does not support it. You cannot use \$LOG.

System Action: Ends the operation.

User Response: To include the logging function in your system, you must do another system generation and include the appropriate supervisor modules. Refer to the *Installation and System Generation Guide* for details on this procedure.

MAXDCB= OUT OF LIMITS

Issued by: \$EDXASM

Explanation: The value specified for the maximum number of chained DCBs is not within the valid range. The valid range of the MAXDCB= operand of the EXIODEV statement is 0–8.

System Action: The error is flagged and compilation continues.

User Response: Specify a value within the range of 0–8 for MAXDCB=.

MAXIMUM NUMBER OF PROGRAMS NOT BETWEEN 1 AND 100

Issued by: \$EDXASM

Explanation: At least one of the MAXPROG= operands on the SYSTEM statement is not within the range of 1–100.

System Action: The error is flagged and compilation continues.

User Response: On the MAXPROG= operand, specify a value between 1–100 for the number of programs which will run concurrently in a partition.

MAX # OF DIRECTORY ENTRIES

Issued by: \$IAMUT1

Explanation: This prompt (using the AL subcommand of DR) requests the maximum number of entries the directory data set is to contain.

System Action: Awaits operator response.

User Response: Enter the maximum number of entries the directory data set is to contain. Enter a null line to use the default of 47 entries. The number of entries can range from 1 to 32767.

MAX # OF DIRECTORY ENTRIES MUST BE AT LEAST xxx

Issued by: \$IAMUT1

Explanation: On a request (using the AL or AS subcommands of DR) to adjust the size of the directory data set, the requested new size is not sufficient to contain all the presently used entries.

System Action: Asks again for the maximum number of directory entries.

User Response: Request a larger maximum number of directory entries.

MAXIMUM OF EIGHT PARMS EXCEEDED

Issued by: \$EDXASM

Explanation: More than 8 parameters were specified for the PARMs= operand.

System Action: The error is flagged and compilation continues.

User Response: Specify between 1 and 8 labels for the PARMs= operand.

MAXIMUM OF 3 PREC= SPECIFICATIONS

Issued by: \$EDXASM

Explanation: For floating-point arithmetic operations, a maximum of three precisions can be specified for the PREC= operand if the RESULT= operand is coded. If RESULT= is not coded, the maximum is two.

System Action: The error is flagged and compilation continues.

User Response: Specify only three precisions for PREC= if RESULT= is coded; otherwise, specify only two precisions.

MAXIMUM OF 8 HEXADECIMAL DIGITS (4 BYTES) PER OPERAND

Issued by: \$EDXASM

Explanation: The operand exceeds the 4-byte maximum.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the operand does not exceed the 4-byte maximum.

MAXIMUM RECORD SIZE xxx

Issued by: \$IAMUT1

Explanation: Using the BF command of DR, this message displays the current maximum record size which can be supported by the Indexed Access Method for any indexed data set with secondary indexes and provides you the opportunity to reset the number.

System Action: If you respond with a null input, no action is taken. If you respond with a new number, the new number is stored and later written to disk as the new maximum record size.

User Response: Reply with a null input to allow the maximum record size to remain as it is. Reply with a new number if you desire to change it. If none of your files has a secondary index, you can specify zero as the size. If you specify a size, the actual amount of storage required to support the specified record size is twice that size plus 8 bytes.

MAXPROG= AND PARTS= DO NOT MATCH

Issued by: \$EDXASM

Explanation: In a SYSTEM statement, the operand MAXPROG must contain the same number of entries as listed in the operand PARTS.

System Action: The error is flagged and compilation continues.

User Response: Make sure the lists coded for MAXPROG and PARTS have the same number of entries.

**MC=YES IS INVALID WITH X.21 TYPE=AC
OR TYPE=DC**

Issued by: \$EDXASM

Explanation: You coded MC=YES on the BSCLINE statement when you specified "AC" (auto-call) or "DC" (direct-call) for the TYPE operand.

System Action: The error is flagged and compilation continues.

User Response: Code MC=NO on the BSCLINE statement if TYPE=AC or TYPE=DC.

**MC=YES NOT VALID WITH
ADAPTER=MFA**

Issued by: \$EDXASM

Explanation: On the BSCLINE statement, MC=YES was specified when ADAPTER=MFA.

System Action: The error is flagged and compilation continues.

User Response: Specify MC=NO when ADAPTER=MFA on the BSCLINE statement.

**MECBLST MUST BE BETWEEN 1 AND 64
(INCLUSIVE)**

Issued by: \$EDXASM

Explanation: During system generation, you coded a value outside the valid range of 1–64 on the MECBLST operand of the SYSTEM statement.

System Action: The error is flagged and compilation continues.

User Response: Code a value from 1–64 for the MECBLST operand and recompile the system definition statements.

MEMBER: xxx UNKNOWN

Issued by: \$UPDATEH

Explanation: The host data set name you specified does not exist and the attempt to open it failed.

System Action: Returns to command mode.

User Response: Specify a valid host data set name and retry the function.

MENUNAME INVALID

Issued by: \$MTM

Explanation: The primary menu name specified for the TERMINAL file record listed immediately before this message is invalid.

System Action: Does not connect the terminal.

User Response: Edit the TERMINAL record and specify the correct primary menu name. Stop and restart the manager.

MESSAGE EXCEEDS 254 CHARACTERS ON LINE NUMBER xxx

Issued by: \$MSGUT1

Explanation: The message in question is too long.

System Action: Continues processing.

User Response: When processing finishes, reduce the length of the message or break it into several messages. Then retry the function.

MISSING OPERAND ON PROC COMMAND - MUST START IN COLUMN 10

Issued by: \$JOBUTIL

Explanation: You specified the job statement data set incorrectly.

System Action: Ends the operation.

User Response: Edit the data set to correct the problem and retry the function.

MESSAGE HAS NOT BEEN ENDED OR CONTINUED ON LINE #xxx

Issued by: \$MSGUT1

Explanation: You did not enter an end character (/*) or continuation character (non-blank in column 71) or on the line in question.

System Action: Continues processing.

User Response: When processing finishes, correct the error and retry.

MISSING OPERAND ON PROGRAM COMMAND - MUST START IN COLUMN 10

Issued by: \$JOBUTIL

Explanation: You specified the job statement data set incorrectly.

System Action: Ends the operation.

User Response: Edit the data set to correct the problem and retry the function.

MISSING OPERAND ON DS COMMAND - MUST START ON COLUMN 10

Issued by: \$JOBUTIL

Explanation: You specified the job statement data set incorrectly.

System Action: Ends the operation.

User Response: Edit the data set to correct the problem and retry the function.

MORE ECB'S SPECIFIED THAN DEFINED BY MAXECB KEYWORD

Issued by: \$EDXASM

Explanation: More ECB labels were specified in the first operand than the MECB has room for.

System Action: Flags the error and continues processing.

User Response: Restrict the number of ECB labels to the number specified the the MAXECB operand.

**MORE THAN EIGHT PARAMETERS
ENCOUNTERED ON LINE #xxx**

Issued by: \$MSGUT1

Explanation: You cannot specify more than eight parameters.

System Action: Continues processing.

User Response: When processing finishes, reduce the number of parameters to eight. Then retry the function.

MORE THAN 16 ADAPTER STATEMENTS

Issued by: \$EDXASM

Explanation: While defining the system configuration, the maximum number of ADAPTER statements was exceeded.

System Action: The error is flagged and compilation continues.

User Response: Specify no more than 16 ADAPTER statements in the system configuration.

**MORE THAN ONE GLOBAL ATTNLIST HAS
BEEN CODED**

Issued by: \$EDXASM

Explanation: Only one global ATTNLIST is allowed per program.

System Action: The error is flagged and compilation continues.

User Response: Specify only one global ATTNLIST in the program.

**MORE THAN ONE LOCAL ATTNLIST HAS
BEEN CODED**

Issued by: \$EDXASM

Explanation: Only one local ATTNLIST is allowed per program.

System Action: The error is flagged and compilation continues.

User Response: Specify only one local ATTNLIST in the program.

**\$MOVEVOL MUST HAVE AT LEAST 512
BYTES OF DYNAMIC STORAGE TO
EXECUTE**

Issued by: \$MOVEVOL

Explanation: There is not enough dynamic storage in the partition to run the utility.

System Action: Ends the operation.

User Response: Allocate more storage to the partition (\$DISKUT2 SS command) and reload the utility.

MSGID= MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: The MSGID= operand parameter specified was not YES or NO, which are the only valid parameters.

System Action: The error is flagged and compilation continues.

User Response: Specify either MSGID=YES or MSGID=NO; the default is MSGID=NO.

M

MSGNO MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The msgno operand was not specified on the MESSAGE instruction and it is a required operand.

System Action: The error is flagged and compilation continues.

User Response: Specify the msgno operand on the MESSAGE instruction as the first operand. It must be a positive integer or the label of an equate preceded by a '+'.

MSGNO MUST BE A POSITIVE SELF-DEFINING TERM OR + EQUATE

Issued by: \$EDXASM

Explanation: The value specified for the msgno operand is not a valid form for that operand.

System Action: The error is flagged and compilation continues.

User Response: Specify the msgno operand as a positive integer or the label of an equate preceded by a '+'.

\$MTMSTORE DATA SET LIMITS EXCEEDED

Issued by: \$MTM

Explanation: The specified \$MTMSTORE file is too small. This can occur after adding a new program with a storage requirement greater than any previous program's requirement, or after adding a new terminal or screen.

System Action: Terminates the Multiple Terminal Manager.

User Response: Delete the \$MTMSTORE file and re-create it with more space.

MULTIPLE TERMINAL MANAGER SYSTEM FAILURE

Issued by: \$MTM

Explanation: The Multiple Terminal Manager task error exit routine has been entered due to a machine or program error.

System Action: The PSW and LSB at the time of failure have been saved at a displacement of X'172' into the program storage. Register 1 in the LSB contains the address of the failing instruction in the case of a program check. The Multiple Terminal Manager program remains active, waiting for an event which will not be posted.

User Response: Use the Event Driven Executive operator facilities to display storage.

MULTIPLE TERMINAL MANAGER TERMINAL FILE RECORDS

Issued by: \$MTM

Explanation: The **TERMINAL** file records processed by the Multiple Terminal Manager are listed after this message. Any messages pertaining to a specific **TERMINAL** file record will be displayed immediately after the file record.

System Action: Lists **TERMINAL** file records processed.

User Response: Review the listing and take action as needed.

N

NAME MUST BE SPECIFIED AND IT MUST BE A LABEL

Issued by: \$EDXASM

Explanation: In the COMP instruction, the name operand was not specified and it is required.

System Action: The error is flagged and compilation continues.

User Response: When the assembly is complete, correct the error and rerun.

NAME: xxx NOT ALLOWED

Issued by: \$UPDATE,\$UPDATEH

Explanation: You specified a name of a data set or volume that is in the wrong format or does not exist.

System Action: Returns to command mode.

User Response: Specify a valid name and retry the function.

NAME NOT ALLOWED

Issued by: \$INITDSK

Explanation: The data set or volume name you specified is restricted to system use, or is a name that the system doesn't recognize.

System Action: Awaits operator response.

User Response: Reenter the information, making sure that the name does not exceed the maximum length (8 characters for data sets, 6 for volumes) and refers to a valid data set or volume.

NAME SUBLIST .GT. PARM SUBLIST

Issued by: \$EDXASM

Explanation: The number of entries in the PARM= field on the USER statement must be greater than or equal to the number of entries specified for the P= field.

System Action: The error is flagged and compilation continues.

User Response: Reduce the number of entries specified for the P= field.

NEITHER -DLE STX- OR -STX- CONTINUING READ

Issued by: \$BSCUT2

Explanation: The utility has not yet received the necessary control characters to indicate the start of transmission of either transparent or non-transparent data. It continues to read from the line.

System Action: Continues processing.

User Response: If the delay in start of transmissions persists, you may want to cancel the current function and retry.

**NEITHER POSITIONAL NOR KEYWORD
PARAMETERS WERE SPECIFIED**

Issued by: \$EDXASM

Explanation: Both positional and keyword operands are missing from the \$IDEF instruction. At least one positional or one keyword operand must be specified for this instruction to be meaningful.

System Action: The error is flagged and compilation continues.

User Response: Specify at least one positional or one keyword operand.

**NETWORK DEACTIVATION IGNORED -
SESSIONS ACTIVE**

Issued by: \$SNAMAIN

Explanation: SNA support has received a request to deactivate the network normally but there are still active sessions in progress.

System Action: SNA support ignores the request to deactivate the network.

User Response: In this situation you have several choices:

- You can deactivate the network from the host, or stop all sessions and have SNA support attempt to deactivate normally again.
- You can use the SNADACT operator command to deactivate the network abnormally.
- You can use the \$C operator command to cancel the SNA application. This should only be used after trying all other possibilities.

If you decide you do not wish to deactivate the network, ignore this message.

NETWORK FUNCTION NETxxx ERROR yyy

Issued by: \$RJESNA

Explanation: An SNA function failed, and \$RJESNA cannot recover from it. The value for xxx indicates the failing SNA function (for example, NETGET, NETPUT, NETINIT). The value for yyy is an SNA return code resulting from the error.

System Action: Terminates \$RJESNA.

User Response: Use this message in conjunction with the "EXTENDED ERROR" message to determine the problem. \$RJESNA requires a two-part support: one native, and one on the host. You may have to consult the programmer who wrote the host interface to help solve this problem. After the problem is corrected, restart \$RJESNA.

NETWORK RE-ACTIVATION STARTED

Issued by: \$SNAMAIN

Explanation: SNA support was either disconnected by the host or detected an unrecoverable hardware error in the network. SNA reconnects with the host because you defined the network with the UNLOAD parameter on the SNAPU statement as either "YES" or "SNADACT." The SNA application reestablished connection with the host.

System Action: SNA support reinitializes its control blocks and then attempts a connection with the host.

User Response: You *may* have to reactivate the network from the host.

NEXTBUF ADDR IS NUMERIC - CAREAD TERMINATED

Issued by: \$\$1ASM

Explanation: The nextbuf operand on the CAREAD instruction is numeric. The first character must be alphabetic.

System Action: Terminates expansion of the CAREAD instruction.

User Response: On the CAREAD instruction, specify a valid nextbuf operand whose first character is alphabetic, and reassemble.

NEXTBUF NAME IS NUMERIC - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The nextbuf operand on the CAREAD instruction is numeric. The first character must be alphabetic.

System Action: Terminates the expansion.

User Response: On the CAREAD instruction, specify a valid nextbuf operand whose first character is alphabetic, and reassemble.

**NEXTBUF NAME TOO LONG - CAREAD
TERMINATED**

Issued by: \$\$1ASM

Explanation: The nextbuf operand on the CAREAD instruction has more than eight characters.

System Action: Terminates expansion of the CAREAD instruction.

User Response: On the CAREAD instruction, specify a nextbuf operand that is less than or equal to eight characters in length, and reassemble.

**NEXTBUF NOT SPECIFIED - EXPANSION
TERMINATED**

Issued by: \$EDXASM

Explanation: The nextbuf operand on the CAREAD instruction is a required operand and is not specified.

System Action: Terminates the expansion.

User Response: On the CAREAD instruction, specify a valid nextbuf operand, and reassemble.

**NLINES MUST BE POSITIVE,
SELF-DEFINING TERM**

Issued by: \$EDXASM

Explanation: The “nline” positional operand (number of values to be printed per line) of the PRINTNUM instruction must be explicitly defined as a positive self-defining term.

System Action: The error is flagged and compilation continues.

User Response: Specify explicitly a positive self-defining term for “nline”.

**NLINES REQUIRED WITH NSPACES
SPECIFICATION**

Issued by: \$EDXASM

Explanation: When the “nspaces” positional operand (spaces between values) is coded on the PRINTNUM instruction, the “nlines” positional operand (number of values to be printed per line) must also be coded.

System Action: The error is flagged and compilation continues.

User Response: Specify the “nlines” operand in addition to the “nspaces” operand.

NO ACTIVE IF OR DO STRUCTURE

Issued by: \$EDXASM

Explanation: No "IF" or "DO" structure is currently being processed.

System Action: The error is flagged and compilation continues.

User Response: Check the "IF" and "DO" logic and ensure an ENDDO, ENDIF, or ELSE was not coded by mistake.

NO ALTERNATE AVAILABLE

Issued by: \$DASDI

Explanation: No alternates remain available for use on the cylinder in use, which contains the RBA in question. A minimum of 7 and a maximum of 14 sectors are available for alternate assignments on a cylinder.

System Action: Continues processing.

User Response: Use another area of the device to store data.

NO BRANCH ADDRESS SPECIFIED

Issued by: \$EDXASM

Explanation: All positional operands on the FIND/FINDNOT instructions are required. This operand specifies the "not found" condition and is the address of the instruction to be executed if the target character is not found.

System Action: The error is flagged and compilation continues.

User Response: Specify the address of the instruction to be executed if the target character is not found.

NO BSC LINE AVAILABLE RJE TERMINATING

Issued by: \$RJE2780, \$RJE3780

Explanation: Either all existing BSC lines are in use, or there are no BSC lines supported by your system. In either case, the utility has no means of sending or receiving data and is unuseable.

System Action: Ends the operation.

User Response: In the case of busy BSC lines, wait and retry the utility. If no BSC lines exist on your system, you cannot use the RJE utility.

**NO BUFFER AVAILABLE TO SEND
REQDISCONT****Issued by:** \$SNAMAIN**Explanation:** SNA support is attempting to deactivate the network. The network cannot be deactivated because there is no SNA buffer to send REQDISCONT to the host.**System Action:** SNA support waits until a buffer is available. This buffer is then used to send REQDISCONT.**User Response:** In this situation you have several choices:

- You can deactivate the network from the host.
- You can wait for SNA support to free a buffer, allowing SNA to send REQDISCONT.
- You can use the \$C operator command to cancel the SNA application. This should only be used after trying all other possibilities.

**NO BUFFER SUPPLIED. \$IAMUT1
TERMINATING****Issued by:** \$IAMUT1**Explanation:** No dynamic storage was specified. Therefore, no buffer exists for \$IAMUT1.**System Action:** Program terminates.**User Response:** Use the SS command of \$DISKUT2 or the third positional operand of the \$L command to specify dynamic storage.**NO BUFF GIVEN - CAWRITE TERMINATED****Issued by:** \$S1ASM**Explanation:** The buffer operand on the CAWRITE instruction is a required operand and none is given.**System Action:** Terminates expansion of the CAWRITE instruction.**User Response:** On the CAWRITE instruction, specify a valid buffer operand, and reassemble.**NO CAIOCB GIVEN - CAxxx TERMINATED****Issued by:** \$S1ASM**Explanation:** The caiocb operand on the CAxxx instruction is a required operand and none is given.**System Action:** Terminates expansion of the CAxxx instruction.**User Response:** On the CAxxx instruction, specify a valid caiocb operand, and reassemble.**NO CHARACTER STRING SPECIFIED****Issued by:** \$EDXASM**Explanation:** All positional operands are required on the FIND/FINDNOT instructions. This operand must specify the address of the string to be searched.**System Action:** The error is flagged and compilation continues.**User Response:** Specify the address of the string to be searched.

**NO COMMUNICATIONS FACILITY
STATIONS STARTED**

Issued by: \$MTM

Explanation: During the initialization process, no Communications Facility stations were able to be started. This message is preceded by another message which more specifically identifies the problem.

System Action: Communications Facility initialization is terminated and Multiple Terminal Manager will not be able to use the Communications Facility.

User Response: Check the message issued prior to this message to determine the problem resolution. Some possible causes of error include DSOPEN error, STATIONS file disk error, initialization work area overflow, no main storage available for CDMCFREC tasks, or Communications Facility not active.

**NO COMMUNICATIONS FACILITY
STATIONS TO BE STARTED**

Issued by: \$MTM

Explanation: The initialization module CDMINIT2 did not find any Communications Facility terminals to be started. If the STATIONS file contained only comments and an end-of-file or the terminals in the STATIONS file were not defined in the TERMINAL file, this message is also issued.

System Action: No Communications Facility stations are started and no asynchronous tasks are loaded. However, Multiple Terminal Manager will be able to operate with all non-Communications Facility functions.

User Response: Ensure that all terminals to be used by Multiple Terminal Manager and Communications Facility are defined as required.

NO DATA IN SUBMIT FILE

Issued by: \$RJESNA

Explanation: The submit function cannot find data in the specified data set.

System Action: Terminates the SUBMIT request.

User Response: Determine if the correct data set was specified and reissue the SUBMIT command.

NO DATA ON SUBMIT FILE

Issued by: \$RJE2780, \$RJE3780

Explanation: There are no commands in the input file data set. The first card is “/*END”.

System Action: Terminates the function. Awaits operator response.

User Response: Correct the data set containing the submit file and resubmit.

NO DS NAME ON /*CONCAT STATEMENT

Issued by: \$RJE2780, \$RJE3780

Explanation: No data set name follows /*CONCAT.

System Action: An /*END card is substituted and the function terminates.

User Response: Add a valid data set name to the /*CONCAT statement.

NO DATA SET NAMES PROVIDED TO \$UPDATE

Issued by: \$UPDATE

Explanation: You ran the utility as a \$JOBUTIL command procedure but did not provide any data set names.

System Action: Ends the operation.

User Response: Edit the job procedure data set to supply the required data set names, and retry the function.

NO DYNAMIC STORAGE ALLOCATED FOR COPY. COPY TERMINATED

Issued by: \$COPYUT1

Explanation: There is not enough dynamic storage available to perform the requested function in the partition use.

System Action: Ends the operation.

User Response: Load the utility into a partition with more storage and retry the function.

NO DISK AT DEVICE ADDRESS xxx

Issued by: \$INITDSK

Explanation: The address you specified refers to a device other than a disk. It is therefore an invalid address for use with \$INITDSK.

System Action: Ends the operation.

User Response: Check for the address of the disk on your system with the \$IOTEST utility. Then retry the \$INITDSK function with a valid device address.

NO GPIB TERMINAL DEFINED IN SUPERVISOR

Issued by: \$GPIBUT1

Explanation: Your system does not recognize or support any GPIB terminals or devices. You cannot use the GPIB utility on your system.

System Action: Ends the operation.

User Response: You can define a GPIB terminal by performing another system generation, and including the required device and supervisor support. Refer to the *Installation and System Generation Guide* for details on this procedure.

NO INPUT PARAMETERS PASSED TO \$PREFIND

Issued by: \$PREFIND

Explanation: You ran the utility in job mode and failed to supply the required parameters.

System Action: Ends the operation.

User Response: Supply the required parameters and retry the function.

NOMSG COMMAND NOT PRECEDED BY PROGRAM COMMAND

Issued by: \$JOBUTIL

Explanation: The procedure statement is in the wrong order.

System Action: Ends the operation.

User Response: Edit the procedure statement and put everything in the correct order. Then retry the function.

NO KEYWORD PARAMETER WAS SPECIFIED FOR THE KEYWORD OPERAND

Issued by: \$EDXASM

Explanation: A keyword operand in this instruction is missing a parameter.

System Action: The error is flagged and compilation continues.

User Response: Code a valid parameter for the keyword and recompile the program.

NO NAME ON /*CONCAT STATEMENT

Issued by: \$RJESNA

Explanation: A /*CONCAT record was encountered in the job stream, but the data set name was missing.

System Action: Terminates submit process as if an I/O error had occurred on the input file.

Cancels the job stream submitted thus far to the host to prevent the host from executing a partial job stream.

User Response: Specify the data set name and resubmit.

NO MATCH ON xxx

Issued by: \$COPYUT1

Explanation: The utility found no match on a copy with a prefix.

System Action: Returns to command mode.

User Response: Check that the prefix is spelled correctly and that the volume in use contains members with that prefix. Retry the function.

**NO NEXTBUF GIVEN - CAREAD
TERMINATED**

Issued by: \$\$S1ASM

Explanation: The nextbuf operand on the CAREAD instruction is a required operand and none is given.

System Action: Does not expand the CAREAD instruction.

User Response: On the CAREAD instruction, specify a valid nextbuf operand, and reassemble.

**NO PARAMETERS PROVIDED TO
\$UPDATE**

Issued by: \$UPDATE

Explanation: You ran the utility in job mode but did not supply the required parameters in the job procedure data set.

System Action: Ends the operation.

User Response: Edit the job procedure data set to correct the problem and retry the function.

**NO OPEN TASK STATEMENT FOR THE
ENDTASK**

Issued by: \$EDXASM

Explanation: There is no TASK statement to correspond to the ENDTASK statement encountered.

System Action: The error is flagged and compilation continues.

User Response: Verify if there should be a TASK statement for this ENDTASK; if not, delete the ENDTASK statement.

**NO OTHER OPERANDS ALLOWED WITH
TIMER, ENTER , OR KEY WAIT**

Issued by: \$EDXASM

Explanation: On the WAIT TIMER, WAIT ENTER, or WAIT KEY instructions, no additional operands may be specified.

System Action: The error is flagged and compilation continues.

User Response: Code the WAIT instruction as explained.

NO PROGRAM LOAD FACILITY**Issued by:** Loader**Explanation:** During initial program load (IPL), it was determined that no program load facility would be available for one of the following reasons:

1. There are no disks configured in the nucleus being IPLed.
2. The resident loader module (RLOADER) was not included in the nucleus.
3. The non-resident loader (\$LOADER) could not be found either on the IPL volume or the fixed-head volume of the IPL device.
4. A disk read/write error (indicated by the previous message) was encountered while searching for \$LOADER.

System Action: The IPL of the system continues with an attempt to attach the task named \$PROG1, which could have been included with the nucleus during system generation.**User Response:** If the ability to load programs is required, make the following corrections as related to the cause above:

1. Add the required DISK statements(s) for a new system generation.
2. Change \$LNKCNTL to INCLUDE RLOADER for a new system generation.
3. Copy \$LOADER from XS4001 to either the IPL volume or the fixed-head volume on the IPL device.
4. Correct the cause of the I/O error in the directory of the IPL volume or fixed-head volume.

NO ROOM TO INSERT**Issued by:** \$IAMUT1**Explanation:** Using the IE subcommand of DR, there is not enough space in the directory data set to insert the entry.**System Action:** Terminates the operation.**User Response:** Either delete unnecessary entries from the directory, or use the AL directory command to enlarge the directory (AS subcommand).**NO SPACE IN DIRECTORY****Issued by:** \$DISKUT1**Explanation:** There is no room to add more members to the volume in use.**System Action:** Returns to command mode.**User Response:** Back up the volume and reallocate it with more directory entries, or delete existing members on the volume.**NO SPACE FOR LOADER/MAX PROG****Issued by:** Loader**Explanation:** Either there is not enough storage to load \$LOADER or the maximum number of programs allowed in the partition has been reached.**System Action:** Terminates \$LOADER**Operator Response:** Change to another partition or regenerate the nucleus with a larger MAXPROG value (see the SYSTEM statement).

NO STORAGE ADDRESS SPECIFIED**Issued by:** \$EDXASM**Explanation:** The required storage address (op2) was omitted for the GETSTORE, PUTSTORE, IPL, or STATUS functions of the TERMCTRL instruction.**System Action:** The error is flagged and compilation continues.**User Response:** Specify the storage address for the TERMCTRL function.**NO STORAGE AVAILABLE FOR CDMINIT2 STATION BLOCK****Issued by:** \$MTM**Explanation:** CDMINIT2 creates a Communications Facility station for itself to receive messages. If CDMINIT2 is unable to create this station, this message is issued.**System Action:** The initialization process terminates. No Communications Facility stations are started; however, Multiple Terminal Manager will be able to operate with all non-Communications Facility functions.**User Response:** To acquire more store in the partition, you can either:

- Selectively cancel non-Multiple Terminal Manager programs active in that partition
- Reduce the number of terminals in the TERMINAL file
- Reduce the size of the Multiple Terminal Manager application area by deleting CDMDUMMY and reallocating it with a smaller size

NO STORAGE AVAILABLE FOR CDMCFREC TASKS**Issued by:** \$MTM**Explanation:** A request for storage to load asynchronous tasks (CDMCFREC) failed.**System Action:** No Communications Facility stations are started and none are usable. Multiple Terminal Manager will be able to operate with all non-Communications Facility functions.**User Response:** To acquire more storage in the partition, you can:

- Selectively cancel non-Multiple Terminal Manager programs active in that partition
- Reduce the number of terminals in the TERMINAL file
- Reduce the size of the Multiple Terminal Manager application area by deleting CDMDUMMY and reallocating it with a smaller size

NO SUCH GPIB DEVICE FOUND**Issued by:** \$GPIBUT1**Explanation:** You specified a GPIB device that the system does not support, or the system does not recognize the name you entered.**System Action:** Ends the operation.**User Response:** Reenter the information, supplying the name of a valid GPIB device.

N

NO SUPERVISOR SPACE IN THIS LIBRARY

Issued by: \$UPDATEH

Explanation: You attempted to update the supervisor but no supervisor space is allocated in the library.

System Action: Returns to command mode.

User Response: Allocate space in the library and retry the function.

NOT ENOUGH SPACE IN TARGET DIRECTORY

Issued by: \$DIUTIL

Explanation: The directory you specified is too small to hold the source directory.

System Action: Returns to command mode.

User Response: Reallocate another data set with more directory space or delete some unneeded members.

NO TAPE FOUND

Issued by: \$TAPEUT1

Explanation: No tape devices are supported by your system, or no tape devices are currently online in your system.

System Action: Ends the operation.

User Response: If tape devices are not turned on, do so. Then retry.

NOT ENOUGH SPACE IN TARGET MEMBER AREA

Issued by: \$DIUTIL

Explanation: The target member of a PDS is too small to hold your data.

System Action: Ends the operation.

User Response: Allocate a larger member in the PDS.

NOT ENOUGH FREE STORAGE TO LOAD THE JOB QUEUE PROCESSOR

Issued by: \$SUBMIT, \$JOBQUT

Explanation: Too many programs are currently executing for the job queue processor to be loaded.

System Action: Ends the operation.

User Response: Determine which programs are running by issuing the \$A ALL command. You may have to wait until a running program terminates, freeing up enough storage for you to load the job queue processor.

NOT ENOUGH STORAGE AVAILABLE FOR DATA PAGING

Issued by: \$IAM

Explanation: Not enough storage was obtained for data paging to be activated - less than 4k of page area was obtained by loading the \$IAMSTGM storage module.

System Action: Data paging is deactivated but \$IAM initialization continues. Any copies of \$IAMSTGM that were loaded are unloaded.

User Response: Check each partition for available storage (use \$A ALL command to see what is loaded in each partition), cancel \$IAM and reload it. Also, make sure you set up your data page area size(s) using the \$IAMUT1 PP command before attempting to load \$IAM.

NOT ENOUGH STORAGE FOR OTHER PROCESSOR

Issued by: \$S1S1UT1

Explanation: The nucleus that the primary processor is using to IPL is larger than the available storage on the secondary processor.

System Action: Returns to command mode.

User Response: Specify a smaller nucleus, or increase storage on the secondary processor.

NO TERMINALS ARE AVAILABLE

Issued by: \$MTM

Explanation: Three possible causes exist: no valid terminal specification records found in the TERMINAL file; no terminal servers can be loaded; all terminals are busy. Other messages generated indicate the problem area.

System Action: Terminates the Multiple Terminal Manager.

User Response: Determine the cause of the problem, and take corrective action.

NO THISBUF GIVEN - CAREAD TERMINATED

Issued by: \$S1ASM

Explanation: The thisbuf operand on the CAREAD instruction is a required operand and none is given.

System Action: Terminates the CAREAD instruction.

User Response: On the CAREAD instruction, specify a valid thisbuf operand, and reassemble.

xxx NOT ON VOL yyy

Issued by: \$RJESNA

Explanation: Series/1 data set xxx could not be found on volume yyy for /*CONCAT, PUNCH, JOURNAL, or SUBMIT processing.

System Action: The function issuing this message ends. In the case of SUBMIT processing, no data is transmitted to the host. In the case of /*CONCAT processing, the job stream is canceled. In the case of PUNCH processing, if a punch data stream is about to be received, the punch function indicates this fact to you and subsequently prompts you for the data set, volume name, and format.

User Response: For punch processing, issue another PUNCH command to define a usable punch file. If a punch data stream is about to be received, you must respond to the system prompts. For SUBMIT or /*CONCAT processing, supply the correct data set or volume name.

NO TRAP CONDITIONS SPECIFIED. \$TRAP TERMINATED

Issued by: \$TRAP

Explanation: No trap conditions have been specified. Some are required.

System Action: Terminates \$TRAP.

User Response: Reload \$TRAP and specify the necessary trap conditions.

NO VOLUME AVAILABLE

Issued by: \$DISKUT2

Explanation: The name you specified does not refer to an existing volume.

System Action: Asks again for the volume name.

User Response: Specify a valid volume and retry the function.

NRF - ID NOT READABLE

Issued by: \$DASDI

Explanation: One or more RBAs have sector IDs which are unreadable.

System Action: Continues processing.

User Response: Assign the RBAs an alternate sector.

NRF - ID NOT VALID

Issued by: \$DASDI

Explanation: One or more RBAs have sector IDs which are not valid.

System Action: Continues processing.

User Response: Assign the RBAs an alternate sector.

**NSPACES MUST BE POSITIVE,
SELF-DEFINING TERM**

Issued by: \$EDXASM

Explanation: The value for “nspaces” (number of spaces between values) on the PRINTNUM instruction must be specified explicitly as a positive self-defining term.

System Action: The error is flagged and compilation continues.

User Response: Specify explicitly a positive self-defining term for “nspaces”.

NUCLEUS NOT FOUND ON VOLUME

Issued by: \$\$S1S1UT1

Explanation: The volume you specified does not contain the nucleus, or you specified the wrong name for the nucleus.

System Action: Returns to command mode.

User Response: Retry the function using the correct nucleus and volume names.

NULL INVALID FOR PARAMETER

Issued by: \$IAMUT1

Explanation: An attempt was made to specify a null response to one or more parameters on which this is invalid.

System Action: Asks again for a parameter value.

User Response: Enter the proper response to the parameter prompt. See the \$IAMUT1 chapter in the IBM Series/1 Event Driven Executive Indexed Access Method (5719-AM3) for a description of each parameter. A null response is only valid for RSVBLK, RSVIX, FPOOL AND DELTHR parameters.

NULL OR INVALID BRANCH TABLE ENTRY

Issued by: \$EDXASM

Explanation: A GOTO branch was improperly coded or formatted.

System Action: The error is flagged and compilation continues.

User Response: Examine the branch and make the needed corrections.

N

NUMBER OF EXTRN/WXTRN SYMBOLS CANNOT EXCEED 255

Issued by: \$EDXASM

Explanation: Your program contains more than 255 EXTRN and WXTRN labels.

System Action: The error is flagged and compilation continues.

User Response: Ensure that your program does not contain more than 255 EXTRN and WXTRN labels.

NUMBER OF FCBS xxx

Issued by: \$IAMUT1

Explanation: Using the BF command, this message displays the current number of FCBs allocated for use in the Indexed Access Method and provides you the opportunity to reset the number.

System Action: If you respond with a null input, no action is taken. If you respond with a new number, the new number is stored and later written to disk as the new number of FCBs.

User Response: Reply with a null input to allow the number of FCBs to remain as it is. Reply with a new number if you desire to change it.

NUMBER OF IACBS xxx

Issued by: \$IAMUT1

Explanation: Using the BF command, this message displays the current number of IACBs allocated for use in the Indexed Access Method and provides you the opportunity to reset the number.

System Action: If you respond with a null input, no action is taken. If you respond with a new number, the new number is stored and later written to disk as the new number of IACBs.

User Response: Reply with a null input to allow the number of IACBs to remain as it is. Reply with a new number if you desire to change it.

NWAIT VALUE GREATER THAN NUMBER OF EVENTS IN OP1 OR MAXECB

Issued by: \$EDXASM

Explanation: The NWAIT operand is coded to wait on more than the listed number of events for this MECB instruction.

System Action: Flags the error and continues processing.

User Response: Change the NWAIT to be equal to or less than the number of events in the list.

OK TO PROCEED?

Issued by: \$IAMUT1

Explanation: Using the LO, UN, or RO commands, this prompt is issued immediately after the message stating that truncation will occur.

System Action: Awaits operator response.

User Response: Reply "Y" to continue processing. Reply "N" to terminate the function.

OMIT THE RECORD AND CONTINUE?

Issued by: \$IAMUT1

Explanation: Using the LO command, an out of sequence or duplicate record in the input data set has been encountered.

System Action: Awaits operator response.

User Response: Reply "Y" to omit the record and continue the function. Reply "N" to terminate the function.

ONE OR MORE UNDEFINED LABELS WERE REFERENCED

Issued by: \$EDXASM

Explanation: Referenced label(s) was not found.

System Action: The error is flagged and compilation continues.

User Response: Examine the name(s) of labels in the listing and check for misspelled or omitted label names. Also examine the undefined and weak external references listed at the end of the compilation.

ONLY ONE NESTING LEVEL ALLOWED

Issued by: \$JOBUTIL

Explanation: The procedure statement is in the wrong order.

System Action: Terminates the utility.

User Response: Edit the procedure statement and put everything in the correct order. Then retry the function.

ONLY ONE OPERAND IS VALID, "LU" OR "HOLDLU"

Issued by: \$EDXASM

Explanation: You coded both LU and HOLDLU on a NETINIT instruction. These operands conflict with each other.

System Action: The error is flagged and compilation continues.

User Response: Code either LU or HOLDLU on the NETINIT instruction.

ONLY 1 HOSTCOMM STATEMENT IS ALLOWED

Issued by: \$EDXASM

Explanation: More than one HOSTCOMM statement was specified in the system configuration.

System Action: The error is flagged and compilation continues.

User Response: Ensure that only one HOSTCOMM statement is specified.

OPEN ERROR xxx

Issued by: \$RJE2780, \$RJE3780

Explanation: An error occurred while attempting to open the BSC line.

System Action: Ends the operation.

User Response: Check the BSC return code to find the cause of the problem, and take appropriate action.

OPEN ERROR FOR xxx

Issued by: \$MTM

Explanation: Either a DSOPEN error occurred on the STATIONS file or on CDMCFREC programs.

System Action: Initialization is terminated (CDMINIT2).

User Response: Normally, this message is issued if the STATIONS file or the CDMCFREC module cannot be found on the PRGMS volume.

OPEN FAILED FOR xxx,yyy

Issued by: \$IAMUT1

Explanation: A \$IAM PROCESS or LOAD request was issued to the indexed file xxx,yyy. The result was an error return code.

System Action: Displays \$IAM return code and terminates the function.

User Response: Examine the Indexed Access Method return code displayed in the subsequent message and take corrective action. Then retry the process.

**OPEN FOR LOAD RETURN CODE = xxx.
RETRY?**

Issued by: \$IAMUT1

Explanation: Using the LO or RO commands, a call to the Indexed Access Method for a LOAD function was unsuccessful. The result was a LOAD return code of xxx.

System Action: Awaits operator response.

User Response: Check the Indexed Access Method return code to find the cause of the problem, and take appropriate action. Reply "Y" to receive the prompt again for data set name and volume. Reply "N" to terminate the function.

**OPEN FOR PROCESS RETURN CODE = xxx.
RETRY?**

Issued by: \$IAMUT1

Explanation: \$IAMUT1 attempted to open the specified Indexed Access Method file in PROCESS mode, and was unsuccessful. The result was a return code of xxx.

System Action: If retry = "Y", asks again for DSNNAME, VOLUME and retries the \$IAM open request. If retry = "N", command terminates.

User Response: Check the Indexed Access Method return code to find the cause of the problem, and take the appropriate action.

OPERAND "ACQUIRE" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the ACQUIRE operand of the NETINIT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code ACQUIRE=YES or ACQUIRE=NO on the NETINIT instruction.

OPERAND "ATTNEV" NOT ALLOWED WITH "FULLDPX=YES"

Issued by: \$EDXASM

Explanation: When the logical unit is in full duplex mode, attention events are not allowed.

System Action: The error is flagged and compilation continues.

User Response: Omit the ATTNEV operand on the NETINIT instruction or code FULLDPX=NO.

OPERAND "BUFF" REQUIRED IF TYPE=RECV/REJECT/LUSTAT

Issued by: \$EDXASM

Explanation: You must code the BUFF operand if you code TYPE=RECV/REJECT/LUSTAT on the NETCTL instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the BUFF operand on the NETCTL instruction.

OPERAND FIELD IS TOO LARGE

Issued by: \$EDXASM

Explanation: The number of operands exceeds the expected maximum for the instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify no more than the maximum number of operands for this instruction.

OPERAND FIELD LENGTH EXCEEDS 254 CHARACTERS

Issued by: \$EDXASM

Explanation: The maximum size of operand parameters for any given instruction is 254 characters.

System Action: The error is flagged and compilation continues.

User Response: Do not code more than 254 characters as the parameter for an operand.

OPERAND "FMH" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the FMH operand of the NETPUT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code FMH=YES or FMH=NO on the NETPUT instruction.

OPERAND "FULLDPX" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the FULLDPX operand of the NETINIT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code FULLDPX=YES or FULLDPX=NO on the NETINIT instruction.

OPERAND "INVITE" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the INVITE operand of the NETPUT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code INVITE=YES or INVITE=NO on the NETPUT instruction.

OPERAND "HOLD" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the HOLD operand of the NETTERM instruction.

System Action: The error is flagged and compilation continues.

User Response: Code HOLD=YES or HOLD=NO on the NETTERM instruction.

OPERAND IS NOT GOTO OR THEN

Issued by: \$EDXASM

Explanation: The "IF" statement must be followed by a GOTO or THEN. If neither is coded, THEN is assumed. Anything else coded is in error.

System Action: The error is flagged and compilation continues.

User Response: Ensure a GOTO or THEN statement follows the IF statement.

OPERAND "HOSTID" IS REQUIRED

Issued by: \$EDXASM

Explanation: The HOSTID operand is required when using the NETINIT instruction.

System Action: The error is flagged and compilation continues.

User Response: Include the HOSTID operand on the NETINIT instruction.

OPERAND "LAST" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the LAST operand of the NETPUT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code LAST=YES or LAST=NO on the NETPUT instruction.

OPERAND "LU" OR "HOLDLU" IS REQUIRED

Issued by: \$EDXASM

Explanation: You did not code the LU operand or the HOLDLU operand on the NETINIT instruction. One is required.

System Action: The error is flagged and compilation continues.

User Response: Code the LU operand or the HOLDLU operand on the instruction.

OPERAND "MSGDATA" REQUIRED IF RESYNC=YES/INIT AND RTYPE=STG

Issued by: \$EDXASM

Explanation: You did not specify the MSGDATA operand on the NETINIT instruction when RESYNC=YES/INIT and RTYPE=STG.

System Action: The error is flagged and compilation continues.

User Response: Code MSGDATA on the NETINIT instruction.

OPERAND NOT VALID WITH WAITM INSTRUCTION

Issued by: \$EDXASM

Explanation: The operand coded is invalid for use with the WAITM instruction.

System Action: The error is flagged and compilation continues.

User Response: Correct the indicated error and reassemble.

OPERAND "RECLen" IS REQUIRED

Issued by: \$EDXASM

Explanation: You must code the RECLen operand when using the NETGET instruction.

System Action: The error is flagged and compilation continues.

User Response: Code the RECLen operand on the NETGET instruction.

OPERAND "RESYNC" NOT ALLOWED WITH "FULLDPX=YES"

Issued by: \$EDXASM

Explanation: When the logical unit is in full duplex mode, you cannot define message resynchronization.

System Action: The error is flagged and compilation continues.

User Response: Code the NETINIT instruction with RESYNC=NO or change to half duplex mode (FULLDPX=NO).

OPERAND "RESYNC" MUST BE "YES," "INIT," OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the RESYNC operand of the NETINIT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code RESYNC=YES, RESYNC=NO, or RESYNC=INIT on the NETINIT instruction.

OPERAND "RTYPE" MUST BE "DISK" OR "STG"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the RTYPE operand of the NETINIT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code RTYPE=DISK or RTYPE=STG on the NETINIT instruction.

OPERAND "VERIFY" MUST BE "YES" OR "NO"

Issued by: \$EDXASM

Explanation: You coded an invalid parameter for the VERIFY operand of the NETPUT instruction.

System Action: The error is flagged and compilation continues.

User Response: Code VERIFY=YES or VERIFY=NO on the NETPUT instruction.

OPERANDS "BUFF" AND "BYTES" REQUIRED FOR "NETGET" AND "NETPUT"

Issued by: \$EDXASM

Explanation: You must code the BUFF and BYTES operands when using the NETGET and NETPUT instructions.

System Action: The error is flagged and compilation continues.

User Response: Code the BUFF and BYTES operands on the NETPUT or NETGET instruction.

OPERAND 1 IS MISSING

Issued by: \$EDXASM

Explanation: The first operand was omitted but must be specified for this instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the first operand for this instruction.

OPERAND 2 IS MISSING

Issued by: \$EDXASM

Explanation: The second operand was omitted but must be specified for this instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the second operand for this instruction.

OPERAND 3 IS MISSING

Issued by: \$EDXASM

Explanation: The third operand was omitted but must be specified for this instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the third operand for this instruction.

OTE TYPE MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The TYPE operand must be specified in an OTE instruction.

System Action: The error is flagged and compilation continues.

User Response: Indicate the type of object element to be defined by coding the TYPE operand.

**OUT OF SEQUENCE, END=YES
PREVIOUSLY SPECIFIED**

Issued by: \$EDXASM

Explanation: END=YES must be coded on the last configuration statement for each device type (terminal,disk) defined during system configuration. In this case, the error occurred when \$EDXASM processed the END=YES for a device type and then encountered another configuration statement for the same device type.

System Action: The error is flagged and compilation continues

User Response: Delete the previous END=YES and make sure that an END=YES is only coded on the last configuration statement for each device type.

**OUT OF SEQUENCE OR DUPLICATE
RECORD**

Issued by: \$IAMUT1

Explanation: Using the LO command, an attempt to load an out of sequence or duplicate record was encountered. The record number message precedes this message and a prompt to continue follows.

System Action: Issues prompt to continue.

User Response: Respond to system prompt.

O

OUT OF SPACE ON PUNCH FILE - ABORTED

Issued by: \$RJESNA

Explanation: The output data set is not large enough to contain the punched output.

System Action: Terminates the output data stream.

User Response: Allocate a larger data set and then retry.

OUT OF SPACE ON PUNCHFILE - PUNCH ABORTED

Issued by: \$RJE2780, \$RJE3780

Explanation: The output data set is too small to hold the punch data.

System Action: Ends the operation.

User Response: Allocate a larger data set and rerun the function.

OUT OF SPACE ON SPOOL FILE

Issued by: \$RJE2780, \$RJE3780

Explanation: The spool file is full.

System Action: Ends the operation.

User Response: Wait until space is available on the file and retry the function.

OUTPUT DATA SET DISK I/O ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the LO or RO commands, a disk read/write error occurred on the output data set. The result was a return code of xxx.

System Action: Current function is terminated.

User Response: Examine the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

OUTPUT DATA SET ERROR. \$IAM RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the LO or RO commands, an Indexed Access Method error occurred on the output data set. The result was a return code of xxx.

System Action: Current function is terminated.

User Response: Examine the Indexed Access Method return code and take appropriate action.

OUTPUT RECORD ASSUMED TO BE xxx BYTES. OK?

Issued by: \$IAMUT1

Explanation: Using the UN command, the output record size is assumed to be the same as the input record size.

System Action: Awaits operator response.

User Response: Reply "Y" to continue processing with the indicated record size. Reply "N" to receive a prompt for the actual record size.

OVERLAY AREA IS TOO SMALL, SIZE REQUIRED IS xxx HEX

Issued by: \$XPSLINK

Explanation: The overlay area you defined on the OVLAREA statement during system generation is not large enough to contain a supervisor overlay segment.

System Action: The error is flagged and the link-edit continues.

User Response: Redefine the starting and ending points of the overlay area to equal size xxx. Link-edit the definition statements with the link control data set.

OVERLAY I/O ERROR

Issued by: \$OVLMGR

Explanation: The overlay manager encountered an I/O error while attempting to read from a disk or diskette.

System Action: The program ends.

User Response: If you are using a diskette unit, make sure the diskette is in the slot and the door is closed. If the diskette is in the slot or if you are reading from a disk, the problem could be a defective record. Use the VD or VV command of the \$INITDSK utility to locate defective records.

PAGE AREA SIZE(S) RESET

Issued by: \$IAMUT1

Explanation: Using the PG command, the new page area size table has been written out to \$IAM.

System Action: When \$IAM is loaded in the future, the new page area sizes will be in effect. Has no effect on the currently executing \$IAM.

User Response: Keep in mind the new page area size the next time you load \$IAM.

PARAMETER OMITTED OR TOO MANY DEVICES SPEC. IN DEVICES= PARAM

Issued by: \$EDXASM

Explanation: The DEVICES= operand must be specified on the ADAPTER statement. Up to four device names can be specified for the MFA, and up to eight device names for the ALPA adapter.

System Action: The error is flagged and compilation continues.

User Response: Specify no more than four device names on the DEVICES= operand.

PARAM COMMAND NOT PRECEDED BY PROGRAM COMMAND

Issued by: \$JOBUTIL

Explanation: The procedure data set is not in the proper order.

System Action: Ends the operation.

User Response: Edit the procedure data set and put everything in the proper order. Then retry the utility.

PARAM POSITION GREATER THAN EIGHT ON LINE #xxx

Issued by: \$MSGUT1

Explanation: The parameter position must be from 1 - 8.

System Action: Continues processing.

User Response: When processing finishes, correct the error and retry.

PARMS= MUST BE A LABEL

Issued by: \$EDXASM

Explanation: One or more of the parameters specified for the PARMS= operand are not labels.

System Action: The error is flagged and compilation continues.

User Response: Specify all parameters for the PARMS= operand as labels.

PARTITION NUMBER IS INVALID

Issued by: \$DUMP

Explanation: An invalid partition number was entered during a partial storage dump.

System Action: Terminates \$DUMP.

User Response: Reload \$DUMP and enter a valid partition number.

PARTITION SIZE EXCEEDS 32 BLOCKS

Issued by: \$EDXASM

Explanation: Entries in the PARTS= operand of the SYSTEM statement (number of 2K blocks per partition) must be within the range of 2–32 (4K–64K).

System Action: The error is flagged and compilation continues.

User Response: Specify a number 2–32 on the PARTS= operand of the SYSTEM statement.

PART NOT ALLOWED WITH OVERLAY PROGRAMS

Issued by: \$EDXASM

Explanation: A cross-partition load (as specified by the PART= operand) is not valid if overlay programs are to be loaded (PGMx operand coded).

System Action: The error is flagged and compilation continues.

User Response: If overlay programs will be loaded, do not specify the PART= operand.

PCB NOT SPECIFIED

Issued by: \$EDXASM

Explanation: The plot control block (PCB) must be specified on the XYPLOT and XTPLOT graphics instructions.

System Action: The error is flagged and compilation continues.

User Response: Specify the label of an 8-word plot control block.

PCI MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: The parameter specified for the PCI= operand of the DCB statement must be either “YES” or “NO”.

System Action: The error is flagged and compilation continues.

User Response: Specifying PCI=YES causes the device to present an interrupt upon completion of the DCB fetch prior to data transfer. PCI=NO will not cause an interrupt to be presented and is the default.

PDEN = COMP NOT VALID ON 4975 MODEL 1

Issued by: \$EDXASM

Explanation: You specified an invalid option for the printer in use.

System Action: Continues processing.

User Response: Retry the function with a valid option.

PERFORMANCE VOLUMES MUST BE ALLOCATED AND THEN IPL SYSTEM AGAIN

Issued by: \$INITDSK

Explanation: While initializing a device, the utility found a performance volume (volume name was specified on the disk system generation statement).

System Action: Returns to command mode.

User Response: When a device is initialized, all performance volumes must be reallocated, or a new system generation must be performed to eliminate the sysgen volumes.

PERMANENT ERROR

Issued by: \$DISKUT2

Explanation: While trying to print a format log, a disk I/O error occurred.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

PERMANENT READ TRANSMISSION ERROR xxx

Issued by: \$RJE2780, \$RJE3780

Explanation: An unrecoverable BSCREAD error occurred. The result was a BSC return code of xxx.

System Action: Ends the utility.

User Response: Check the BSC return code to find the cause of the problem, and take appropriate action.

PERMANENT WRITE TRANSMISSION ERROR xxx

Issued by: \$RJE2780, \$RJE3780

Explanation: An unrecoverable BSCWRITE error occurred, resulting in a BSC return code of xxx.

System Action: Ends the utility.

User Response: Check the BSC return code to find the cause of the problem, and take appropriate action.

POINTERS IN HEADER OF HIGH INDEX BLOCK NOT ZERO

Issued by: \$VERIFY

Explanation: In the highest level index block, the upward pointer, forward pointer, and backward pointer should all be zero. In this case, they were not.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt. File may have to be regenerated.

POLL ADDRESS MUST BE 2 HEX DIGITS

Issued by: \$EDXASM

Explanation: The poll address was specified incorrectly. A valid poll address consists of two hexadecimal digits.

System Action: The error is flagged and compilation continues.

User Response: Specify the poll address as two hexadecimal digits for the POLL operand.

POLL= KEYWORD NECESSARY AND VALID ONLY IF TYPE=MT

Issued by: \$EDXASM

Explanation: The POLL= operand is required if ADAPTER=MFA and TYPE=MT are specified on the BSCLINE statement.

System Action: The error is flagged and compilation continues.

User Response: Specify a list of 1–4 poll/select addresses as the POLL operand.

PORT= AND SECADDR= MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The PORT= and SECADDR= operands were not specified for the ADAPTER statement, and they are required.

System Action: The error is flagged and the compilation continues.

User Response: Specify the PORT= and SECADDR= operands and reassemble.

PORT NOT NUMERIC - CAIOCB TERMINATED

Issued by: \$EDXASM

Explanation: The PORT= operand on the CAIOCB statement is not numeric.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement specify a valid PORT= operand that is a decimal number between zero and 31, and reassemble.

PORT NUMBER INVALID

Issued by: \$EDXASM

Explanation: On the ADAPTER statement, the port number specified is not in the correct format or range.

System Action: The error is flagged and the compilation continues.

User Response: Code the PORT= operand with the value 0 or 1 and reassemble.

**PORT NUMBER TOO LARGE - CAIOCB
TERMINATED**

Issued by: \$EDXASM, \$S1ASM

Explanation: The PORT= operand on the CAIOCB statement contains a value greater than 31.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a PORT= operand that is less than or equal to 31, and reassemble.

**PORT NUMBER TOO SMALL - CAIOCB
TERMINATED**

Issued by: \$EDXASM, \$S1ASM

Explanation: The PORT= operand on the CAIOCB statement contains a value less than zero.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a PORT= operand that is greater than or equal to zero, and reassemble.

**PORT WAS NOT SUPPLIED - CAIOCB
TERMINATED**

Issued by: \$EDXASM, \$S1ASM

Explanation: The PORT= operand on the CAIOCB statement is required but was not supplied.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a valid PORT= operand, and reassemble.

**PORTS NUMBER TOO LARGE -
EXPANSION TERMINATED**

Issued by: \$EDXASM

Explanation: The PORTS= operand on the CACBx statement contains a value greater than 31.

System Action: Terminates the expansion.

User Response: On the CACBx statement, specify a PORTS= operand that is less than or equal to 31, and reassemble.

**PORTS NUMBER TOO SMALL -
EXPANSION TERMINATED**

Issued by: \$EDXASM

Explanation: The PORTS= operand on the CACBx statement contains a value less than zero.

System Action: Terminates the expansion.

User Response: On the CACBx statement, specify a PORTS= operand that is greater than or equal to zero, and reassemble.

**PREVIOUSLY DEFINED FORMAT
DESTROYED**

Issued by: \$IMAGE

Explanation: Due to an open error the previously defined format is gone.

System Action: Ends the operation.

User Response: Redo the edit and retry the function.

**PRIMARY MENU xxx FAILED FOR
TERMINAL yyy**

Issued by: \$MTM

Explanation: A SETPAN function for the terminal yyy using the primary menu xxx has failed.

System Action: Does not include the terminal in the Multiple Terminal Manager system.

User Response: Ensure that a valid menu name is specified in the TERMINAL file for the specified terminal.

**PRIME INDEX DATASET CONTAINS NO
RECORDS**

Issued by: \$IAMUT1

Explanation: During the secondary index load, no primary records were located in the prime file. Therefore no records can be loaded into the secondary file.

System Action: Control returns to \$IAMUT1 main procedure.

User Response: Load the prime file first, and retry the operation.

**PRINT ERROR - ATTEMPT CORRECTION
THEN RESPOND BELOW**

Issued by: \$RJESNA

Explanation: The Event Driven Executive has detected a printer problem (printer was powered off). A subsequent message, "ENTER C(CONTINUE), T(TERMINATE), R(REDIRECT):" is issued to allow for recovery from this error.

System Action: Suspends printing until the problem is corrected.

User Response: Retry the operation by entering a "C", or cancel the PRINT command by entering a "T". If a "T" is entered, the host is notified that the print function is not available, and a command to the host is required to start the printer. To redirect the output to a different printer, specify "R".

PROCESSING /*CONCAT - FILE ERROR xxx

Issued by: \$RJESNA

Explanation: \$RJESNA encountered a disk error while opening the data set specified on the /*CONCAT record. The result was a return code of xxx.

System Action: Cancels the job stream to prevent a partial job stream from executing on the host. Processing is terminated as if an error condition had occurred on the input file.

User Response: Check the disk and diskette read/write return codes to find the cause of the problem, and take the appropriate action.

PROC COMMAND NOT ALLOWED AFTER PROGRAM COMMAND

Issued by: \$JOBUTIL

Explanation: The procedure statement is in the wrong order.

System Action: Ends the operation.

User Response: Edit the procedure statement and put everything in the correct order. Then retry the function.

PROGRAM AREA TOO SMALL TO HOLD PGM xxx

Issued by: \$MTM

Explanation: The manager's program area is too small to hold the named program.

System Action: Does not use the program xxx.

User Response: Increase the program area size by reallocating CDMDUMMY, or split the program into smaller link-edited programs.

PROGRAM AREA - NOT RELEASED

Issued by: \$STGUT1

Explanation: The utility inspected the area and found it to be a program, which cannot be released with \$STGUT1.

System Action: Returns to command mode.

User Response: If you still want to release the area of storage, use the \$C operator command or re-IPL the system.

PROGRAM CANNOT BE TRACE FILE

Issued by: \$BSCUT1

Explanation: You specified that a program type data set be used to hold BSC trace data. Only a data type data set can be used as a trace file.

System Action: Ends the operation.

User Response: Specify a data type data set for use as the trace file. If necessary, allocate such a data set.

PROGRAM CAPACITY EXCEEDED

Issued by: \$VERIFY

Explanation: The amount of working storage allocated to \$VERIFY is insufficient to process the indexed file specified.

System Action: \$VERIFY terminates.

User Response: Increase the amount of working storage available to \$VERIFY. Refer to the IBM Series/1 Event Driven Executive Indexed Access Method (5719-AM3) for a description of how to calculate the amount of working storage required and how to modify the amount supplied.

**PROGRAM FILE LARGER THAN PROGRAM
MANAGER BUFFER**

Issued by: \$MTM

Explanation: The program table built during initialization exceeds the size of the buffer used by the program manager.

System Action: Multiple Terminal Manager terminates.

User Response: Increase the program manager buffer size in module CDMCOMMN, assemble and relink \$MTM.

PROGRAM LOAD ERROR

Issued by: \$MTM

Explanation: An Event Driven Executive LOAD error occurred for the requested program.

System Action: Does not load the program.

User Response: Determine the cause of the problem. Rebuild the program if the problem persists.

PROTECT = MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: The PROTECT= operand must be specified as either yes or no.

System Action: Continues processing.

User Response: After the assembly completes, correct the error and rerun.

PUNCH ABORTED - FILE ERROR xxx

Issued by: \$RJESNA

Explanation: An error occurred while a punch stream was being received from the host and written to a Series/1 data set, or an I/O error occurred during the opening of the data set. The result was a return code of xxx.

System Action: None.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**PUNCH DATA BEING RECEIVED - NO
PUNCH FILE DEFINED**

Issued by: \$RJESNA

Explanation: A punched output data set is about to be received from the host, but no punch data set has been defined to \$RJESNA.

System Action: Issues a prompt requesting the punch data set name.

User Response: Enter the punch data set name.

**PUNCH DATA BEING RECEIVED - NO
PUNCH FILE DEFINED**

Issued by: \$RJE2780, \$RJE3780

Explanation: There is no punch file defined to hold the punch data being received from the host.

System Action: Prompts for definition of a punch file: "ENTER PUNCH FORMAT - S OR O:".

User Response: Answer S for source, or O for object. Note: DS2 on program load is used.

**P1= INVALID WHEN USING CAPS= WITH
IMMEDIATE DATA**

Issued by: \$EDXASM

Explanation: P1= cannot be used on the PRINTTEXT or READTEXT instruction if the CAPS= character count is specified. P1= can be used if CAPS=YES is specified instead.

System Action: The error is flagged and compilation continues.

User Response: Specify either CAPS= character count and omit P1, or specify CAPS=YES with P1 coded.

QUERY MESSAGE MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The required prompt message was omitted on the QUESTION instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify the prompt message on the QUESTION instruction.

R

RANGE= MUST BE HIGH FOR MFA ADAPTER

Issued by: \$EDXASM

Explanation: RANGE=LOW cannot be specified if ADAPTER=MFA on the ACCA TERMINAL statement.

System Action: The error is flagged and compilation continues.

User Response: Specify RANGE=HIGH. If omitted, RANGE=HIGH is the default.

RAN OUT OF MAIN STORAGE IN PARTITION

Issued by: \$MTM

Explanation: Using the MTMUTIL utility, there was insufficient storage in the partition to load either the program or to contain the loader.

System Action: The requested program is not loaded.

User Response: Specify a different partition to MTMUTIL.

RBA xxx, ALT UNASSIGNMENT - DATA ERR

Issued by: \$DASDI

Explanation: During an alternate unassignment, the data field was moved, but an uncorrectable Error Condition Code (ECC) was detected. This indicates that the integrity of the data in the unassigned alternate has been compromised. The xxx is the RBA in question.

System Action: Continues processing.

User Response: Refresh the data.

RBA xxx, ALTERNATE ASSIGNED - DATA ERROR

Issued by: \$DASDI

Explanation: During an alternate assignment, the data field was moved, but an uncorrectable Error Condition Code (ECC) was detected. The ECC will be inverted, indicating that the integrity of the data in the assigned alternate has been compromised. The xxx is the RBA in question.

System Action: Continues processing.

User Response: Refresh the assigned alternate.

RBA ERROR: xxx BAD FLAG - NO ALTERNATE FOUND

Issued by: \$DASDI

Explanation: The RBA in question, xxx, has an unreadable sector ID.

System Action: Continues processing.

User Response: Assign the RBA an alternate sector.

RBA xxx SECTOR WRITTEN - DATA ERR

Issued by: \$DASDI

Explanation: An RBA has returned a data check to the read data. The integrity of the data in this RBA has been compromised.

System Action: Continues processing.

User Response: Refresh the RBA.

RBN xxx CONTAINS INVALID BACKWARD POINTER.

Issued by: \$VERIFY

Explanation: The indicated index block did not properly point to the previous index block on the same level.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt.

RBN xxx CONTAINS INVALID FORWARD POINTER.

Issued by: \$VERIFY

Explanation: The indicated data block did not properly point to the next data block on the same level.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt.

RBN xxx CONTAINS INVALID UPWARD POINTER.

Issued by: \$VERIFY

Explanation: The indicated file block did not properly point to its higher level index block.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt.

RBN xxx IS IN FREEPOOL CHAIN, BUT IS NOT A VALID FREEPOOL BLOCK.

Issued by: \$VERIFY

Explanation: Every block in the freepool chain should have a block type of X'8000'. The indicated block did not.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt.

**"RDSCB" REQUIRED IF
RESYNC=YES/INIT AND RTYPE=DISK**

Issued by: \$EDXASM

Explanation: You did not specify the RDSCB operand on the NETINIT instruction when RESYNC=YES/INIT and RTYPE=DISK.

System Action: The error is flagged and compilation continues.

User Response: : Disk resynchronization requires an open dataset control block (DSCB). Code the RDSCB operand on the NETINIT instruction to point to the DSCB.

**READ ERROR DURING READBACK CHECK
OF TARGET DATASET**

Issued by: \$COPYUT1

Explanation: A read error occurred in the target data set.

System Action: Ends the operation.

User Response: Choose a different target data set and retry the function.

**READ ERROR ENCOUNTERED ON xxx.yyy
RC = zzz**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: \$XPSPRE was unable to read the specified data set. The result was a return code of zzz.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

READ ERROR (OBJECT MODULE)

Issued by: \$UPDATE

Explanation: A read error was encountered on the data set name.

System Action: Returns to command mode.

User Response: Recreate the object module and retry the function.

READ ERROR OCCURRED, RC = xxx

Issued by: \$XPSLINK from \$XPSPOST

Explanation: When attempting to read a record from disk or diskette, an error occurred. The result was a return code of xxx.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. Rerun \$XPSLINK.

**READ ERROR ON RECORD NUMBER xxx
RC = yyy**

Issued by: \$COPY

Explanation: An error occurred while reading the disk record xxx. The result was a return code of yyy.

System Action: Returns to command mode.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

READ ERROR ON SOURCE , RC = xxx

Issued by: \$TAPEUT1

Explanation: The source device used for the copy detected a read error.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**READ ERROR ON TAPE - RC = xxx
DUMP ENDED**

Issued by: \$TAPEUT1

Explanation: A read error occurred that caused the dump to terminate.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**READ ERROR - TAPE RC = xxx
TAPE RECORD NUMBER = yyy**

Issued by: \$TAPEUT1

Explanation: An error occurred while reading the tape record yyy.

System Action: Returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**READ INPUT DATASET RETURN CODE =
xxx. RECORD NUMBER = yyy**

Issued by: \$IAMUT1

Explanation: An attempt to read the indexed input data set failed for a sequential data set. The result was a return code of xxx which occurred at record yyy.

System Action: Ends the command.

User Response: For LO command, check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. For the UN or RO commands, check the Indexed Access Method return code to find the cause of the problem, and take appropriate action.

R

READ INPUT DATASET RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: Using the UN or RO commands, a call to the Indexed Access Method for a GETSEQ function was unsuccessful. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the Indexed Access Method return code to find the cause of the problem, and take appropriate action.

REASSIGN UNSUCCESSFUL - TERMINAL CODE xxx

Issued by: \$TERMUT1

Explanation: The attempt to reassign the address of the terminal failed.

System Action: Returns to command mode.

User Response: Check the terminal I/O general return code to find the cause of the problem, and take appropriate action.

RECONNECT SYNTAX INVALID

Issued by: \$MTM

Explanation: The correct syntax has not been used on the RECONNECT operator command.

System Action: Ignores the command.

User Response: Retry the RECONNECT command with correct syntax.

RECONNECT TERMINAL DEFINITION ERROR

Issued by: \$MTM

Explanation: The RECONNECT operator interface facility has encountered a failure while attempting to reconnect a terminal to the Multiple Terminal Manager. Since initialization already would have performed all functions necessary to include the terminal in the terminal table, the TERMINAL file, SCRNS volume or source table in RECONNED has probably been altered since the Multiple Terminal Manager was started.

System Action: Does not connect terminal.

Explanation: Determine the cause of the error (check TERMINAL file for correct data).

RECORD DOES NOT VERIFY INPUT RECORD = xxx CHECK VALUE = yyy

Issued by: \$TAPEUT1

Explanation: There is a discrepancy between the record the utility expected to read and the record it actually read.

System Action: Returns to command mode.

User Response: Enter a new command if necessary.

**RECORD MATCH NOT FOUND FOR
SECONDARY INDEX**

Issued by: \$VERIFY

Explanation: While verifying cross references, the program failed to find a match for a specific secondary or primary key.

System Action: Issues messages that identify the primary and secondary key values and continues processing.

User Response: Secondary file does not match primary file and should be rebuilt.

RECORD NUMBER = xxx

Issued by: \$IAMUT1

Explanation: Using the LO, UN, or RO commands, the record number being processed is identified when an error occurs. This message is preceded by the error message.

System Action: Dependent on the preceding error message.

User Response: Examine the preceding error message and take corrective action.

**RECORD OUT OF SEQUENCE NEAR RBN
xxx**

Issued by: \$VERIFY

Explanation: Records in the indicated data block were out of sequence. The records should be in ascending order by key.

System Action: Verification continues.

User Response: In most cases the IAM file must be rebuilt.

RECORD SIZE MUST BE NUMERIC

Issued by: \$IAMUT1

Explanation: Using the LO or UN commands, your reply to the prompt ENTER RECORD SIZE was not numeric.

System Action: Reissues the prompt for record size.

User Response: Enter a numeric value for the record size.

REGISTER SPECIFICATION INVALID

Issued by: \$EDXASM

Explanation: Index registers #1 and #2 cannot be specified directly in the instruction. The registers can be used, however, in the parameter format: (parameter, #reg).

System Action: The error is flagged and compilation continues.

User Response: Recode the instruction.

**RELATIVE RECORD OUT OF RANGE
ALLOWABLE RECORDS ARE
0001 - xxx HEX**

Issued by: \$DISKUT2

Explanation: The relative record number you entered while using the PR or LR command of the \$DISKUT2 utility is invalid.

System Action: The system asks for a relative record number within the allowable range.

User Response: Enter a relative record number within the range displayed in the message.

**REMOTE MANAGEMENT UTILITY ERROR
2 - COMMUNICATIONS OPEN FAILED,
RETURN CODE: xxx**

Issued by: \$RMU

Explanation: The OPEN of the BSC communication line failed. The result was a BSC return code of xxx.

System Action: Terminates the program.

User Response: Check the BSC return code to find the cause of the problem, and take the appropriate action.

**REMOTE MANAGEMENT UTILITY ERROR
1 - INSUFFICIENT BUFFER. SIZE: xxx**

Issued by: \$RMU

Explanation: The size of the buffer defined for use by the utility is xxx bytes, which is less than the 512-byte minimum. The default 1024-byte buffer size has been modified incorrectly.

System Action: Terminates the program.

User Response: Check the buffer size, respecify a valid size, and restart the program.

**REMOTE MANAGEMENT UTILITY ERROR
3 - COMMUNICATIONS CLOSE FAILED,
RETURN CODE: xxx**

Issued by: \$RMU

Explanation: The CLOSE of the BSC communication line failed. The result was a BSC return code of xxx.

System Action: Terminates the program.

User Response: Check the BSC return code to find the cause of the problem and take the appropriate action.

**REMOTE MANAGEMENT UTILITY ERROR
4 - COMMUNICATIONS I/O ERROR. I/O
FUNCTION xxx RETURN CODE: yyy**

Issued by: \$RMU

Explanation: BSCAM encountered an error during an I/O function, xxx, which can be one of the following:

- Read initial
- Read continue
- Write EOT
- Write initial
- Write EOT (ABORT)
- Write continue

The result was a BSC return code of yyy.

System Action: In cases of either a time-out error (return code 10) or a sequence error (return code 12), the system terminates the current function and waits for the next request from the host. In cases of other error types, the system terminates after reporting the I/O error.

User Response: Check the BSC return code to find the cause of the problem and take the appropriate action.

**REMOTE MANAGEMENT UTILITY ERROR
5 - LOAD OVERLAY FAILED, RETURN
CODE: xxx OVERLAY NUMBER yyy**

Issued by: \$RMU

Explanation: The utility attempted to load an overlay program yyy with a LOAD instruction. The load failed, resulting in a LOAD return code of xxx.

System Action: Terminates the program.

User Response: Check the LOAD return code to find the cause of the problem, and take the appropriate action.

**REMOTE MANAGEMENT UTILITY ERROR
6 - OVERLAY FUNCTION MISSING.
FUNCTION: xxx OVERLAY NUMBER yyy**

Issued by: \$RMU

Explanation: The utility's function table erroneously defined a function xxx as being contained within overlay yyy.

System Action: Terminates the program.

User Response: Check that the user written function is added properly to the function table. Correct any problem and restart the program.

R

REMOTE MANAGEMENT UTILITY ERROR 7 - INVALID MULTIPOINT TRIBUTARY SELECT

Issued by: \$RMU

Explanation: The host system has improperly sent the utility a multipoint "select" when it should have sent a "poll".

System Action: The utility returns an EOT (end of transmission) to the host, terminates the function in progress, and reads for the next request.

User Response: Determine the problem with the host program that is communicating with the utility, correct it, and rerun the program.

REMOTE MANAGER BUSY - REMOTE RECORDS NOT STORED

Issued by: \$LOG

Explanation: The \$LOG utility is unable to get access to the Remote Manager's buffers in order to write the log records. This message indicates a problem with the IBM-supplied software.

System Action: \$LOG is active but it cannot write log records to the Remote Manager's buffers.

User Response: Contact your local IBM service representative.

REMOTIQUE OVERFLOW - REMOTE RECORD NOT STORED

Issued by: \$LOG

Explanation: The Remote Manager's buffers cannot handle the number of log records being written by the \$LOG utility.

System Action: \$LOG continues to write log records but the records are not stored for the Remote Manager.

User Response: Refer to the description manual for the RM1 (Remote Manager) program product in order to correct this problem.

RENAME ERROR. RETURN CODE = xxx

Issued by: \$IAMUT1

Explanation: During the process of adjusting the size of the directory data set using the AS subcommand of DR, the rename of the new data set from \$WRKDIR to \$IAMDIR failed. The result was a return code of xxx.

System Action: Displays another message indicating the temporary name and volume of the new directory data set.

User Response: Check the system return code to find the cause of the problem, and take appropriate action.

REPLACEMENT OF PROGRAM xxx WAS NOT SPECIFIED

Issued by: \$UPDATE

Explanation: You attempted to update an existing program but did not specify that the new update replace the old one.

System Action: Ends the operation.

User Response: Specify replacement of the existing program and retry.

REQUIRED KEYWORD IS MISSING

Issued by: \$EDXASM

Explanation: For the STORBLK instruction, the TWOKBLK operand must be coded.

System Action: Continues processing but does not generate object text for that instruction.

User Response: Correct indicated error and reassemble.

REQUIRED PARAMETER IS MISSING

Issued by: \$EDXASM

Explanation: The positional operands required for PUTEDIT and GETEDIT instructions are "text", "(list)", and either "format" or "(format,list)".

System Action: The error is flagged and compilation continues.

User Response: Ensure the required positional operands are specified as explained.

REQUIRED STORAGE EXCEEDS 63K. NOT SET.

Issued by: \$IAMUT1

Explanation: Using the BF command, the total amount of storage required for the requested buffer size, number of IACBs, number of FCBs and maximum record size exceeds 63K bytes. It would be impossible to load \$IAM into a single partition with this amount of dynamic storage.

System Action: Ends the operation.

User Response: Specify more realistic values for the requested parameters.

RESET OPERAND DOES NOT ALLOW ADDITIONAL OPERANDS

Issued by: \$EDXASM

Explanation: For the STIMER instruction, if the first operand is coded RESET, no additional operands may be coded.

System Action: Continues processing but does not generate object text for that instruction.

User Response: Correct the indicated error and reassemble.

R

RESET TYPE NOT SPECIFIED OR INVALID ASSUME RESET PUNCH? (Y,N)

Issued by: \$RJESNA

Explanation: A RESET command or a RESET command with an undefined parameter was entered.

System Action: Awaits operator response.

User Response: Answer "Y" or "N" depending on whether you wish to reset the punch data set name and format. If "N" is entered, the message "RESET NOT DONE" is issued and the RESET command is cancelled. If "Y" is specified, \$RJESNA resets the punch function.

RESOURCE BUSY

Issued by: Spooling

Explanation: A \$\$ command has been issued before a previous command has finished processing.

System Action: The second \$\$ command is ignored.

Operator Response: Allow the first \$\$ command to complete, then reissue the command.

RESTART ON; CD/DS/GS/MA/MS/SO COMMANDS INVALID

Issued by: \$SPLUT1

Explanation: After the spool facility has been restarted, the utility cannot perform any of the other listed command functions.

System Action: Returns to command mode.

User Response: Do not attempt to enter any of the listed commands.

RESULT= MAY NOT BE SPECIFIED WITH MOVE OR MOVEA

Issued by: \$EDXASM

Explanation: The "RESULT=" operand is invalid when used with MOVE or MOVEA instructions. The result of the operation is stored in the first operand by the MOVE and MOVEA instructions.

System Action: The error is flagged and compilation continues.

User Response: Remove the "RESULT=" operand from the MOVE and MOVEA instruction.

RESULT= OPERAND MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The RESULT= operand must be specified in an arithmetic operation as it has been explicitly implied in the PREC= operand.

System Action: The error is flagged and compilation continues.

User Response: Code the RESULT= operand with a variable.

**RETRY SUPPLIED IS NOT NUMERIC -
EXPANSION TERMINATED**

Issued by: \$EDXASM

Explanation: The RETRY= operand on the CACB1 statement is not numeric.

System Action: Terminates the expansion.

User Response: On the CACB1 statement, specify a valid RETRY= operand that is numeric, and reassemble.

RETRY SUPPLIED IS NOT NUMERIC

Issued by: \$S1ASM

Explanation: The RETRY= operand on the CACB1 statement is not numeric.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CACB1 statement, specify a valid RETRY= operand that is numeric, and reassemble.

**\$RJESNA SESSION TERMINATED BY
HOST**

Issued by: \$RJESNA

Explanation: \$RJESNA received a -16 return code (session terminated by host) in response to an SNA NETGET command.

System Action: \$RJESNA terminates.

User Response: If you entered a host session termination command, or the host operator terminated the session, no response is required. If session termination was not caused by either of these methods, you may have to consult the host systems programmer to determine the cause of session termination.

\$RJESNAM FILE ERROR xxx

Issued by: \$RJESNA

Explanation: \$RJESNA encountered an error while trying to read from its message data set \$RJESNAM. The result was a return code of xxx.

System Action: Terminates \$RJESNA.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

R

RSB= MUST BE EVEN

Issued by: \$EDXASM

Explanation: The number specified for the number of residual status bytes on the EXIODEV statement must be zero or an even decimal number between 4 and 16 inclusive.

System Action: The error is flagged and compilation continues.

User Response: Specify a zero or an even decimal number from 4–16 for RSB=.

RSB= OUT OF LIMITS

Issued by: \$EDXASM

Explanation: The number specified for the number of residual status bytes on the EXIODEV statement must be zero or an even decimal number between 4 and 16 inclusive.

System Action: The error is flagged and compilation continues.

User Response: Specify a zero or an even decimal number from 4–16 for RSB=.

R TYPE MUST BE ON MFA BASE ADDRESS

Issued by: \$EDXASM

Explanation: 4975-01R and 4975-02R type devices must be attached to the MFA base address. This address is specified on the MFA ADAPTER statement.

System Action: The error is flagged and compilation continues.

User Response: Specify the MFA base address on the ADDRESS= operand of the TERMINAL statement.

SCAN= PARAMETER IS INCORRECT

Issued by: \$EDXASM

Explanation: The parameter for the SCAN= operand of the GETEDIT/PUTEDIT instruction is specified incorrectly. Valid parameters are "FIXED" and "FREE".

System Action: The error is flagged and compilation continues.

User Response: Specify either "FIXED" (input data adheres to format statement) or "FREE" (input data is separated by a blank, comma, or slash).

SCREEN TABLE LARGER THAN INPUT BUFFER

Issued by: \$MTM

Explanation: The screen table built during initialization exceeds the Input Buffer size.

System Action: Aborts initialization.

User Response: Increase the Input Buffer size in module CDMCOMMN, reassemble and relink \$MTM.

SE= MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: The parameter specified for the SE= operand of the DCB statement must be either "YES" or "NO".

System Action: The error is flagged and compilation continues.

User Response: Specifying SE=YES will suppress exception condition reporting. SE=NO allows exception reporting and is the default.

SECONDARY ADDRESS OUT OF RANGE

Issued by: \$EDXASM

Explanation: The value in the SECADDR= operand was out of the acceptable range.

System Action: The error is flagged and the compilation continues.

User Response: Specify a value within the range 0 - 255 for the SECADDR= operand and reassemble.

S

SECONDARY INDEX DATASET OPEN ERROR, \$DISKUT3 RTCODE= xxx

Issued by: \$IAMUT1

Explanation: During the secondary index load, \$IAMUT1 was unable to open the secondary index data set as specified. The result was a \$DISKUT3 return code of xxx.

System Action: System prompts you: RETRY (Y/N)?

User Response: Check the \$DISKUT3 return code to find the cause of the problem and respond to the prompt accordingly.

SECOND DIGIT OF ADDR FIELD NOT HEX

Issued by: \$S1ASM

Explanation: The device address operand on the CAxxx instruction has an invalid hexadecimal character as the second digit.

System Action: Terminates expansion of the CAxxx instruction.

User Response: On the CAxxx instruction, specify a valid device address operand whose second digit is a hexadecimal character, and reassemble.

SECOND DIGIT OF PORT IS NOT NUMERIC

Issued by: \$S1ASM

Explanation: The PORT= operand on the CAIOCB statement has a non-numeric character as the second digit.

System Action: Terminates expansion of the CAIOCB statement.

User Response: On the CAIOCB statement, specify a valid PORT= operand whose second digit is numeric, and reassemble.

SECOND OPERAND NOT RESET OR CLEAR

Issued by: \$EDXASM

Explanation: On the WAIT ECB instruction, the second operand specified can only be RESET or CLEAR.

System Action: The error is flagged and compilation continues.

User Response: Specify as the second operand (if coded) either RESET or CLEAR.

SELF-DEFINING TERM INVALID FOR PDEN/PMODE/CHARSET KEYWORDS

Issued by: \$EDXASM

Explanation: If the 4975 character set operands PDEN, PMODE, or CHARSET are specified, a self-defining term cannot be coded.

System Action: The error is flagged and compilation continues.

User Response: If these operands are specified, code either a parameter name, an indexed value (register), or an address.

SENSOR I/O DEVICE AT ADDRESS xxx IS OFFLINE BSCA NOT THE DEVICE AT ADDR: yyy

Issued by: Sensor I/O

Explanation: The system checks the status of any defined sensor I/O or Binary Synchronous Communications Adapter devices, and prints appropriate status messages.

System Action: None.

User Response: This is an informational message. No user response is needed.

SEQUENTIAL RECSIZE GT BLOCKSIZE

Issued by: \$IAMUT1

Explanation: Using the LO or UN commands, a response to a prompt for sequential data set block size was less than the record size. This is an invalid block size.

System Action: The block size prompt is reissued.

User Response: Specify the correct block size.

SESSIONS RESET - LINK LEVEL ACTIVE

Issued by: \$SNAMAIN

Explanation: The host detected a serious error with the network and is reinitiating network activation. The host restarts SDLC by sending a SDLC SNRM command.

System Action: SNA support clears the system of all sessions and resets all its control blocks for SDLC and SNA.

User Response: You can restart lost sessions by reissuing NETINIT instructions. You can find out what error the host encountered by using the host network problem determination utilities.

S

SHOULD THE 3101 DATA STREAM BE SAVED?

Issued by: \$IMAGE

Explanation: The utility is asking if you want to save the edited screen for use on a 3101 terminal.

System Action: Awaits operator response.

User Response: Answer "yes" to save the 3101 image, and "no" to save only the 4978/4979 image.

SIGNON PROGRAM NOT AVAILABLE FOR TERMINAL xxx

Issued by: \$MTM

Explanation: The specified terminal is required to sign on and off but no program named SIGNON was found in the PRGRMS volume.

System Action: Does not include the terminal.

User Response: Place a program named SIGNON in the PRGRMS file, or designate that no signon is needed for the specified terminal. Restart the Multiple Terminal Manager.

S1S1 ATTACHMENT SPECIFIED IS INOPERATIVE

Issued by: \$\$S1S1UT1

Explanation: The utility could not obtain status from the specified attachment and therefore marks it inoperative.

System Action: Ends the operation.

User Response: Check the attachment for hardware problem. Then retry the function.

SIZE OF TARGET DIRECTORY DOES NOT MATCH SIZE OF SOURCE DIRECTORY

Issued by: \$MOVEVOL

Explanation: In order for the requested function to take place, the sizes of both directories must match.

System Action: Ends the operation.

User Response: Retry the function using directories that match in size. Or, using the original directories, delete and reallocate one so that its size matches the other.

\$SNA LOAD ERROR xxx

Issued by: \$RJESNA

Explanation: A problem was encountered by \$RJESNA during an attempt to load SNA. The result was a LOAD return code of xxx.

System Action: Terminates \$RJESNA.

User Response: Check the LOAD return code to find the cause of the problem, and take the appropriate action.

Note: The Event Driven Executive SNA must be on the same volume as \$RJESNA if \$SNA is loaded by \$RJESNA.

**SNA/SDLC CONFIGURATION ERROR,
RETURN CODE = xxx**

Issued by: \$SNAMAIN

Explanation: SNA support detected a configuration error. The return code xxx indicates the type of error.

System Action: SNA support ends all sessions and either unloads or attempts to reactivate depending on how you previously defined the UNLOAD operand on the SNAPU statement.

User Response: Check the return code description for the cause of the error. The return codes are described under "SDLC Communications Return Codes" on page MC-325.

**SNA/SDLC IS MANUAL CALL - PLEASE
DIAL YOUR NETWORK EXTENSION**

Issued by: \$SNAMAIN

Explanation: The network has been loaded and is waiting for the host connection.

System Action: None.

User Response: Dial the network telephone number.

**SNA/SDLC TERMINATED DUE TO A
PERMANENT SDLC ERROR CONDITION
SDLC DEVICE ERROR - RETURN CODE
= xxx**

Issued by: \$SNAMAIN

Explanation: SNA support encountered an unrecoverable hardware error. The return code xxx indicates the type of error.

System Action: SNA support ends all sessions and either unloads or attempts to reactivate depending on how you previously defined the UNLOAD operand on the SNAPU statement.

User Response: Check the return code description for the cause of the error. The return codes are described under "SDLC Communications Return Codes" on page MC-325.

**SNA SYSTEM FAILURE - SNA TERMINATED
DUE TO UNRECOVERABLE ERROR
CONDITIONS**

Issued by: \$SNAMAIN

Explanation: \$SNAMAIN either program checked or encountered an unrecoverable error condition.

System Action: The system deactivates the network and unloads \$SNAMAIN and \$SNA. It then dumps the Level Status Block (LSB), Task Control Block (TCB), and the SNA control blocks and buffers for \$SNAMAIN.

User Response: Take a stand-alone dump of storage and perform problem determination.

SNA UNLOADING - NO SEND BUFFERS WERE SPECIFIED FOR BUFFER POOLING

Issued by: \$SNAMAIN

Explanation: You defined the network incorrectly causing SNA to unload. You must define some send buffers when using Buffer Pooling (BUFPOOL=YES on the SNAPU statement). Send buffers are equal to SBUFNO on the SNAPU statement plus the SENDBUF on all the LU statements.

System Action: SNA unloads.

User Response: Recode the network parameter statements to include send buffers and then regenerate the network.

SOFT ERR - ECC CORRECTED

Issued by: \$DASDI

Explanation: One or more RBAs were Error Condition Code corrected. These RBAs will have decreased performance time, but the data is still good.

System Action: Continues processing.

User Response: To improve performance time, refresh the sectors. The integrity of the data will not be compromised. If problems persist and use of the sectors does not decrease performance time, you can assign alternate sectors.

\$\$SORTL LOAD ERROR, LOAD RTCODE=xxx

Issued by: \$IAMUT1

Explanation: During the secondary index load, \$IAMUT1 was unable to load the \$\$SORTL program. This resulted in a return code of xxx.

System Action: Ends the operation.

User Response: Check the LOAD return code to find the cause of the problem, and take appropriate action. This error can be caused by insufficient storage available in the current partition. Try a larger partition.

SORT001P, SORT/MERGE SPECIFICATION PHASE STARTED

Issued by: Sort/Merge

Explanation: The specification phase of Sort/Merge has begun. Specification statements will be syntax checked for errors.

System Action: Continues processing.

User Response: This is an informational message. No user response is needed.

SORT002T, ODD BLOCK SIZE INVALID FOR TAPE

Issued by: Sort/Merge

Explanation: The Data Set Definition specification statement for tape has indicated an odd block size for tape. Odd blocks are not valid for tape.

System Action: Sort/Merge terminates the job.

User Response: Change the block size to an even number and restart the job.

SORT004W, TERMINAL DEVICE NAME INVALID; INVOKING TERMINAL ASSUMED

Issued by: Sort/Merge

Explanation: The terminal device name on the log statement (positions 3-10) is invalid. It contains leading blanks.

System Action: Sort/Merge assumes the invoking terminal will be used as the terminal device. If the warning message option (position 14 of the header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: Specify a device name starting in positions 3-10 which has been defined on a TERMINAL statement at system generation. Resubmit the job.

SORT006T, HEADER STATEMENT MISSING

Issued by: Sort/Merge

Explanation: No header statement was found.

System Action: The job ends.

User Response: Include missing header statement and resubmit the job.

SORT007T, JOB TYPE MISSING OR INVALID, POSITIONS 2-7

Issued by: Sort/Merge

Explanation: A job type (SORTA, SORTR, SORTRS or MERGE) was not specified in the header statement.

System Action: The job ends with no additional specification processing.

User Response: Specify the job type in positions 2-7 of the header statement and resubmit the job.

SORT008S, CONTROL FIELD LENGTH MISSING OR INVALID, 256 ASSUMED

Issued by: Sort/Merge

Explanation: The control field length (positions 8-10 in the header statement) is in error for one of the following reasons:

- No control field length was specified.
- There is either a nonnumeric character or an embedded blank in positions 8-10.
- The control field specified is greater than 256 or less than 1.

System Action: Sort/Merge assumes a control field length of 256 and continues the specification phase. The job ends after the specification phase.

User Response: Place a valid number in positions 8-10 of the header statement and resubmit the job.

SORT009S, ASCENDING/DESCENDING OPTION INVALID, POS 11; A ASSUMED

Issued by: Sort/Merge

Explanation: Position 11 of the header statement has neither a blank or an A for ascending, nor a D for descending sequence.

System Action: Sort/Merge assumes ascending sequence and continues the specification phase. The job ends after the specification phase.

User Response: Enter an A or D in position 11 of the header statement and resubmit the job.

SORT010S, INVALID DSD SPECIFICATION, POS 2; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: The referenced statement is a DSD statement. It should have an I (for input), O (for output) or a W (for work) in position 2, however, it does not.

System Action: Sort/Merge bypasses the statement and goes on to the next statement. The job ends after the specification phase.

User Response: Specify an I, O, or W in position 2 of the referenced statement and resubmit the job.

SORT011S, WARNING MESSAGE OPTION INVALID, POSITION 14; T ASSUMED

Issued by: Sort/Merge

Explanation: The warning message option on the header statement (position 14) is not blank, T or P.

System Action: Sort/Merge assumes a T in position 14, indicating termination of the specification phase if warning messages exist. The job ends after the specification phase.

User Response: Specify either a blank, T, or P in position 14 of the header statement and resubmit the job.

**SORT012W, PRINT OPTION INVALID;
ASSUMED TO BE 0**

Issued by: Sort/Merge

Explanation: The print option in either the referenced header statement (position 12) or the referenced log statement (position 2) is not blank, 0, 1, or 2.

System Action: Sort/Merge assumes a 0 for the print option. If the warning message option (position 14 of the header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: Specify a blank, 0, 1, or 2 as the print option and resubmit the job.

**SORT014W, I/O ERROR OPTION IS
INVALID, POSITION 17; T IS ASSUMED**

Issued by: Sort/Merge

Explanation: Position 17 of your header statement has neither a blank, T, S or A to indicate I/O error options.

System Action: Sort/Merge assumes a T in position 17, indicating termination on I/O error conditions. If the warning message option (position 14 of the header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: Enter a blank, T, S or A in position 17 of the header statement and resubmit the job.

**SORT015S, HEADER DROP/KEEP OPTION
INVALID, POS 13; DROP ASSUMED**

Issued by: Sort/Merge

Explanation: Position 13 of the header statement is neither a blank, a K, or a D.

System Action: Sort/Merge assumes a D in position 13 (drop control field) and continues the specification phase. The job ends after the specification phase.

User Response: Enter a blank, a K, or a D in position 13 of the header statement and resubmit the job.

**SORT016W, COLLATING OPTION INVALID;
EBCDIC ASSUMED - POSITION 16**

Issued by: Sort/Merge

Explanation: The collating option in position 16 of the header statement is invalid. It is neither blank, E or A.

System Action: Sort/Merge assumes an E in position 16 (EBCDIC) to indicate collating sequence. If the warning message option specifies continue processing (a P in position 14 of the header statement), Sort/Merge continues processing after the specification phase. If it is blank or T, the job ends after the specification phase.

User Response: Enter a blank, E, or A in position 16 of the header statement to indicate collating sequence. Resubmit the job.

S

SORT017S, OUTPUT RECORD LENGTH > HEADER CONTROL FIELD LENGTH

Issued by: Sort/Merge

Explanation: This is a SORTR, SORTRS or MERGE job. Position 13 of the header statement indicates the control field is not being dropped. The output record length must include the control field length.

System Action: A length equal to the output record length plus the largest total control field (positions 8-10) is assumed. The specification phase continues processing. The job ends after the specification phase.

User Response: Recompute your output record length, and change the record size of your output data set. Resubmit the job.

SORT018S, INPUT DS ORGANIZATION OPTION INVALID, POS 15; BLANK ASSUMED

Issued by: Sort/Merge

Explanation: The entry in position 15 of the header statement for this job is invalid for one of the following reasons:

- This is a SORTA job which requires a consecutive data set organization. Position 15 does not contain a blank or C indicating such an organization.
- This is a SORTR, SORTRS, or MERGE job and position 15 does not contain a valid entry, which must be blank or C for a consecutive data set organization, or an I for an indexed organization.

System Action: Sort/Merge assumes a consecutive data set organization and continues processing. The job ends after the specification phase.

User Response: Enter a valid character (blank, C, or I in position 15 of the header statement) and resubmit the job.

SORT019T, INVALID RECORD SIZE FOR xxx

Issued by: Sort/Merge

Explanation: After issuing a CALL to request the EXTRACT function of the Indexed Access Method, for the data set xxx, Sort/Merge found the record size to be zero.

System Action: Sort/Merge terminates processing.

User Response: Position 15 of the header statements indicates that the data set is an Indexed Access Method data set. The data set probably contains all zeroes, indicating it has not been defined or loaded as an Indexed Access Method data set. Correct the error and resubmit the job.

SORT020T, ILLOGICAL ACTION CODE - NO OUTPUT DATA SET - SORT TERMINATED

Issued by: Sort/Merge

Explanation: The action code returned by the output exit routine indicates that Sort/Merge should take an action that is either impossible or illogical, or both. No output data set exists for Sort/Merge to continue processing the sorted records.

System Action: Sort/Merge terminates the job.

User Response: Check the logic of your routine in providing the action code.

SORT021T, RETURN CODE ISSUED xxx yyy zzz

Issued by: Sort/Merge

Explanation: Sort/Merge is processing the data set zzz. After issuing a CALL to request the yyy function, Sort/Merge received a return code of xxx; yyy can be one of the following:

Instr.	Function requested
<i>IAM-PROC</i>	Process (PROCESS)
<i>IAM-EXTR</i>	Extract (EXTRACT)
<i>IAM-GET</i>	Get (GET)
<i>IAM-DISC</i>	Disconnect (DISCONN)

System Action: Sort/Merge terminates processing for all of the above functions except GET, when the user I/O error exit routine, when provided, determines the disposition of the error. Otherwise, Sort/Merge continues processing as specified by the I/O error option in position 17 of the header statement.

User Response: Check the Indexed Access Method return code for an explanation of the error. If a user error occurred, correct it and resubmit the job.

S

SORT022T, NO SPECIFICATIONS PROVIDED

Issued by: Sort/Merge

Explanation: If a specification data set was provided, no specifications were found in it.

If Sort/Merge was initiated from a user program with the LOAD instruction, and specifications were provided in main storage, the number of specification statements indicated was less than 1.

System Action: Sort/Merge terminates the job.

User Response: If a specification data set was provided, be sure it contains the specifications. If Sort/Merge was initiated from a user program, and specifications are provided in main storage, be sure the number of specifications indicated is greater than 0.

SORT023S, SORTA JOB, INPUT RECORD NOT FIXED

Issued by: Sort/Merge

Explanation: This is a SORTA job. The record format of the input data set must be fixed.

System Action: Sort/Merge ends after the specification phase.

User Response: Provide an input data set with a record format of fixed, and resubmit the job.

SORT024S, OUTPUT RECORD LENGTH LESS THAN 4 - SORTA JOB

Issued by: Sort/Merge

Explanation: This is a SORTA job. The record length defined for the output data set is incorrect. It must be equal to 4 or greater than 4.

System Action: Sort/Merge continues processing using an output record length of 4. The job ends after the specification phase.

User Response: Recreate the output data set with a record length of 4 or more bytes.

SORT025S, INVALID SPECIFICATION TYPE; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: The referenced statement cannot be processed. It is not a comment, DSD, ALTSEQ, include/omit or field statement.

System Action: Sort/Merge bypasses the statement. The job ends after the specification phase.

User Response: Correct the statement or remove it from the specifications. Resubmit the job.

**SORT026S, MERGE JOB, OUTPUT RECORD
FOR JOB > OUTPUT DATA SET
RECSIZE**

Issued by: Sort/Merge

Explanation: This is a MERGE job. The length of the output record created during the job is greater than the record length defined for the output data set. If you drop the control field (position 13 of the header statement contains a D), the length of the output record is equal to the record length (RECSIZE) defined for the input data set.

If you keep the control field (position 13 of the header statement contains a K or blank), the length of the output record is equal to the length of the control field (in positions 8-10 of the header statement) plus the record length (RECSIZE) defined for the input data set.

If the control field is dropped, Sort/Merge assumes the record length (RECSIZE) defined for the output data set to be equal to the record length defined for the input data set.

If the control field is kept, Sort/Merge assumes the record length defined for the output data set to be equal to the length of the input record plus the total control field length (specified in position 8-10 of the header statement).

System Action: The job ends after the specification phase.

User Response: Determine the correct length of your output record and correct the output DSD statement. Resubmit the job.

**SORT027S, FR FIELD IN SET; SUMMARY
FIELD MAY NOT FOLLOW; FD
ASSUMED**

Issued by: Sort/Merge

Explanation: This a SORTRS job with an input data set of variable record format. The logical order of the referenced summary field statement (FS in positions 1-2) is invalid, because an entire record as a data field (FR in positions 1-2 of a field statement) has previously been specified in this include set.

System Action: The referenced summary field statement is treated as a delimited data field statement (FD in positions 1-2). The job ends after the specification phase.

User Response: Either remove the statement or change its order, and resubmit the job.

**SORT028S, ALTSEQ STATEMENT HAS
INVALID DATA**

Issued by: Sort/Merge

Explanation: The data to alter the normal sequence in the referenced ALTSEQ statement is in error. One of the 4-position groups (positions 3-6, 7-10, etc.) does not contain a valid hexadecimal number (0-9 or A-F).

System Action: The job ends after the specification phase.

User Response: Correct the statement and resubmit the job.

**SORT029T, INCLUDE OR OMIT
STATEMENT INVALID FOR MERGE JOB**

Issued by: Sort/Merge

Explanation: This is a MERGE job. Include and omit statements are not valid specifications for a MERGE job. The referenced statement was either an include or an omit.

System Action: The job ends.

User Response: Check to make sure that a MERGE job was wanted. If it was, remove all include and/or omit statements. Verify that any remaining specification statements are correct and resubmit the job.

**SORT031S, INVALID INCLUDE/OMIT
CONTINUATION; RECORD BYPASSED**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR, or SORTRS job. The referenced statement is part of an include or omit set. It should have an A (AND) or O (OR) in position 2. However, it does not.

System Action: Sort/Merge bypasses the statement and goes on to the next statement. The job ends after the specification phase.

User Response: Specify an A or O in position 2 of the referenced statement and resubmit the job.

**SORT030S, FIRST INCLUDE/OMIT OF SET
SPECIFIES CONTINUATION**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is the first in an omit or include set. However, it contains a non-blank character in position 2. A blank was expected.

System Action: Sort/Merge ignores the non-blank character. The job ends after the specification phase.

User Response: Enter a blank in position 2 of the referenced statement and resubmit the job.

**SORT032S, AND/OR NOT SPECIFIED,
POSITION 2; OR IS ASSUMED**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is part of an include or omit set. It should contain an A (AND) or O (OR) in position 2; however, position 2 is blank.

System Action: Sort/Merge assumes an O in position 2 and continues the specification phase. The job ends after the specification phase.

User Response: Check the order of your statements. Determine if the referenced statement should have an A or an O in position 2 and resubmit the job.

**SORT033S, FACTOR 1 LENGTH EXCEEDS
64; 64 IS ASSUMED**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is an include or omit statement for character data (position 3 contains a C). Factor 2 is a field (position 14 contains an F). The length of factor 1 exceeds 64 bytes.

System Action: Sort/Merge assumes a factor 1 length of 64 and continues the specification phase. The job ends after specification phase.

User Response: Ensure that the factor 1 and factor 2 field location are correct. If they are (the field exceeds 64), use multiple include statements (specifying an A for AND in position 2) to cover the length of the entire field.

**SORT034S, FACTOR 2 CONSTANT;
INVALID DATA OR LENGTH**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is an include or omit statement containing a factor 2 constant specified in positions 15-34. The factor 2 constant is in error for one of the following reasons:

- The length of the factor 2 constant is not the same length as the factor 1 field specified in positions 4-11.
- The data of the factor 2 constant is not consistent with the data type specified in position 3.

System Action: Sort/Merge continues the specification phase. The job ends after the specification phase.

User Response: Ensure that the factor 2 constant is consistent with the factor 1 field and the data type field specified in position 3. Resubmit the job.

S

SORT035S, CHARACTER CONSTANT - LENGTH EXCEEDS 20; 20 IS ASSUMED

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is an include or omit statement for character data (position 3 contains a C). Factor 2 is a character constant (position 14 contains a C). The length of factor 1 is greater than 20, which is the maximum length allowed.

System Action: Sort/Merge assumes a length of 20 for factor 1 and continues the specification phase. The job ends after the specification phase.

User Response: Specify a length of 20 or less in positions 4-11 of the referenced statement and resubmit the job.

SORT036S, INVALID RELATIONAL, POSITIONS 12-13

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is an include/omit statement and positions 12-13 (the relational between factor 1 and factor 2) does not contain an EQ, NE, LT, GT, LE or GE.

System Action: Sort/Merge continues the specification phase. The job ends after the specification phase.

User Response: Correct the statement and the resubmit job.

SORT037S, ZONE SPECIFIED - RELATIONAL NOT EQ OR NE

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. A zoned data type has been specified in position 3 of the referenced include/omit statement and positions 12-13 do not contain EQ or NE.

System Action: Sort/Merge continues the specification phase. The job ends after the specification phase.

User Response: Specify an EQ or NE in positions 12-13 of the referenced statement, and resubmit the job.

SORT038S, INVALID FACTOR 2 TYPE, POSITION 14; C IS ASSUMED

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR, SORTRS job. The referenced statement is an include/omit statement and position 14 contains neither an F (field) nor C (constant).

System Action: Sort/Merge assumes factor 2 is a constant (C in position 14) and continues the specification phase. The job ends after the specification phase.

User Response: Specify a C or an F in position 14 of the referenced statement and resubmit the job.

**SORT039S, INVALID FIELD TYPE
SPECIFICATION, POS 2; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: The referenced statement is a field specification; however, position 2 does not contain an F, N, O, D, S, or R.

System Action: The referenced statement is bypassed. Processing continues with the next statement. The job ends after the specification phase.

User Response: Enter an F, N, O, D, S, or R in position 2 and resubmit the job.

**SORT040W, DATA SPECIFICATION IN
SORTA/MERGE JOB; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: This is a SORTA or MERGE job. A data field specification (positions 1 and 2 contain FD, FR or FS) is not valid for a SORTA or MERGE job.

System Action: Data field specifications are treated as comment cards and are bypassed. Processing continues with the next statement. If the warning message option (position 14 of the header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: Check to ensure that a SORTA or MERGE job was wanted. If so, remove all data specifications and resubmit the job.

**SORT041W, SUMMARY SPECIFICATION
FOUND, NOT SORTRS JOB; FD
ASSUMED**

Issued by: Sort/Merge

Explanation: This is a SORTR job. A summary data field specification (positions 1 and 2 contain FS) is valid only for SORTRS jobs.

System Action: All FS specifications are treated as data specifications (an FD in positions 1-2). If the warning message option (position 14 of header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: If a SORTR job is wanted, remove all summary specifications before running the job.

SORT042S, FIELD STATEMENT NOT IN LOGICAL ORDER; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: This is a SORTR or SORTRS job. Control field statements (FN, FO or FF in positions 1-2) must precede data field statements (FD, FR or FS in positions 1-2) in an include set.

The logical order of the referenced statement is invalid for one of the following reasons:

- The referenced statement is a control field specification which has been preceded by a data field specification.
- The referenced statement is a data field specification which has not been preceded by a control field specification.

System Action: The statement is bypassed. Processing continues with the next statement. The job ends after the specification phase.

User Response: This statement is either misplaced or incorrectly specified. Correct the statement and resubmit the job.

SORT043W, LOCATION FIELD NONBLANK, POSITIONS 4-11; BLANK ASSUMED

Issued by: Sort/Merge

Explanation: The location field (position 4-11) in the referenced field statement contains non-blanks. The referenced field statement is one of the following:

- Summary field statement with FSV in positions 1-3.
- Forced data character statement with FDV in positions 1-3.
- Unconditional force statement.
- Force-otherwise statement.
- Conditional force continuation statement.

The location field (positions 4-11) contains non-blanks.

System Action: If the warning message option (position 14 of the header statement) is P, the Sort/Merge operation completes. If it is blank or T, the job ends after the specification phase.

User Response: Specify blanks in positions 4-11 and resubmit the job.

**SORT044W, CONDITIONAL FORCE; POS
4-7 NONBLANK; BLANK ASSUMED**

Issued by: Sort/Merge

Explanation: The referenced statement is a conditional force field statement. The location information in position 4-7 of the referenced statement is invalid. This field should be blank, but contains non-blanks.

System Action: The location information in position 4-7 is ignored and blanks are assumed. Sort/Merge continues to process the statement. If the warning message option (position 14 of the header statement) is P, Sort/Merge continues processing after the specification phase; if the option is blank or T, the job ends after the specification phase.

User Response: Specify blanks in position 4-7 and resubmit the job.

**SORT045S, CONDITIONAL FORCE
LOCATION INVALID; POS 8-11; 1
ASSUMED**

Issued by: Sort/Merge

Explanation: The referenced statement is a conditional force field statement. The location information in position 8-11 of the referenced statement is invalid for one of the following reasons:

- The field contains a nonnumeric character or embedded blanks.
- The field entry is not right justified.
- The location is greater than 4096.
- The location is less than 1.

System Action: The location field is ignored and the length is assumed to be 1. Sort/Merge continues processing. The job ends after the specification phase.

User Response: Correct the location field (positions 8-11) and resubmit the job.

S

SORT046S, ILLEGAL CONDITIONAL CONTINUATION; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: The referenced statement is a conditional continuation statement (FF in positions 1-2, non-blank in positions 12 and 14). However, it does not follow a conditional force specification.

System Action: Sort/Merge bypasses the statement. The job ends after the specification phase.

User Response: Remove the statement, or precede it with a conditional force statement. Resubmit the job.

SORT047S, ILLEGAL FORCE-OTHERWISE CONTINUATION; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: The referenced statement is a field statement for a force-otherwise specification (F in position 2, blank in position 12, non-blank in position 14). However, it does not follow a conditional force specification.

System Action: Sort/Merge bypasses the statement. The job ends after the specification phase.

User Response: Remove the statement, or precede it with a conditional force statement. Resubmit the job.

SORT048S, EXTENDED SUMMARY FIELD LENGTH INVALID; LENGTH BYPASSED

Issued by: Sort/Merge

Explanation: This is a SORTRS job. The referenced statement is a summary data field (FS in positions 1-2). The length of the extended summary field (positions 15-16) is invalid for one of the following reasons:

- The field contains an invalid number or embedded blanks.
- The length is:
 - Greater than 18 if D or U specified in position 3
 - Greater than 8 if P specified in position 3
 - Not 2, 4 or 8 if S specified in position 3
- The length is less than 1.
- The entry is not right justified to position 16.
- The length is less than or equal to the summary data field length.

System Action: Sort/Merge assumes that no entry was made in positions 15-16 and continues the specification phase. The job ends after the specification phase.

User Response: Correct the entry in positions 15-16 and resubmit the job.

SORT049S, DATA TYPE INVALID OR MISSING, POSITION 3; C IS ASSUMED

Issued by: Sort/Merge

Explanation: The reference statement contains an invalid or missing data type (C/Z/D/P/U/S/V/A/L/T) in position 3.

System Action: Sort/Merge assumes a C in position 3 and continues the specification phase. The job ends after the specification phase.

User Response: Correct the data type in position 3 and resubmit the job.

SORT050S, DATA TYPE INVALID OR MISSING, POSITION 3; D IS ASSUMED

Issued by: Sort/Merge

Explanation: The referenced summary field statement (positions 1-2 contain FS) contains an invalid or missing data type in position 3. The valid data types are D/P/U/S or V.

System Action: Sort/Merge assumes a D in position 3 and continues the specification phase. The job ends after the specifications phase.

User Response: Correct the data type in position 3 and resubmit the job.

SORT051S, POS 4-11 OR POS 15-22 ARE INVALID; LENGTH 1 IS ASSUMED

Issued by: Sort/Merge

Explanation: The referenced statement is either an include/omit statement (I or O in position 1) or a field statement (FN, FO, FD, or FS only in positions 1-2).

The location information in positions 4-11 and/or positions 15-22 of the referenced statement is invalid for one of these reasons:

- The **from** or **to** location contains a non-numeric character or embedded blanks
- The **from** or **to** entry is not right justified
- The **to** location is blank
- The **from** or **to** location is zero
- The **from** location is greater than the **to** location
- The lengths of factor 1 and factor 2 are not the same
- The **from** or **to** location is greater than 4096

System Action: The location field(s) is ignored for this statement and the length is assumed to be 1. Sort/Merge continues processing. The job ends after the specification phase.

User Response: Correct this statement and resubmit the job.

SORT052S, TO OR FROM FIELD NOT WITHIN INPUT RECORD

Issued by: Sort/Merge

Explanation: In the referenced include/omit or field statement, either or both entries in the **to** or **from** positions contain a value larger than your input record length. If record format is variable, the value is larger than the maximum record length of the data set, minus 4 bytes for the segment descriptor.

System Action: The location field(s) is ignored. The job ends after the specification phase.

User Response: Ensure that you have the desired input data set. Also check that the **from** and **to** positions contain the desired values. Correct statement and the resubmit job.

SORT053S, DATA LENGTH INVALID FOR DATA TYPE; MAXIMUM ASSUMED

Issued by: Sort/Merge

Explanation: In the referenced field or include/omit statement, the length of the data field is invalid for the data type in position 3, as follows:

<i>Data Type</i>	<i>Valid Length</i>
C	1-64
Z	1
D	1-18
U	1-18
P	1-8
S	2, 4, or 8
A	1-18
L	2-19
T	2-19

System Action: Sort/Merge assumes the maximum length for the data type specified in position 3. The job ends after the specification phase.

User Response: Check the from and to positions of the referenced statement. Make the necessary corrections and resubmit the job.

**SORT054S, NO SUMMARIZATION IN
FIRST INCLUDE SET; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: This is a SORTRS job with multiple include sets. The first include set did not specify a summary field, however the referenced statement in the current include set does (FS in positions 1 and 2). If summarization of data is desired, all include sets must have summary field statements.

System Action: Sort/Merge bypasses the referenced summary statement and all subsequent summary statements. Sort/Merge ends after the specification phase.

User Response: Determine if summarization is desired. If not, remove the summary statement(s) from the specifications. If so, specify summarization for all include sets. Resubmit the job.

**SORT055S, SUMMARY FIELD BYPASSED;
DIFFERS FROM FIRST INCLUDE SET**

Issued by: Sort/Merge

Explanation: This is a SORTRS job with multiple include sets. The referenced statement specifies a summary field (FS in position 1-2). Compare the referenced field statement with the field statement(s) that follow the first include set.

System Action: Sort/Merge bypasses the referenced summary statement and all subsequent summary statements in this include set. Sort/Merge ends after the specification phase.

User Response: Ensure that the summary specifications for each include set agree with the summary specifications for the first include set regarding number, length, location and data type of summary fields and regarding the summary overflow indicator (V).

**SORT056S, SUMMARY DATA FIELDS IN
INCLUDE SET > 24; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: This is a SORTRS job. More than 24 summary data fields (FS in positions 1 and 2) have been specified for the current include set.

System Action: The referenced summary statement is bypassed and Sort/Merge continues the specification phase. The job ends after the specification phase.

User Response: Limit the number of summary data fields to 24 or less for each include set and resubmit the job.

SORT057S, MORE THAN ONE OVERFLOW INDICATOR IN INCLUDE SET; FDV ASSUMED

Issued by: Sort/Merge

Explanation: This is a SORTRS job. The referenced field statement specifies a summary field overflow indicator (FSV in positions 1-3). At least one other overflow indicator has already been specified for this include set.

System Action: The referenced statement is treated as a forced data character (FDV in positions 1-3); the job ends after the specification phase.

User Response: Change the extra summary field overflow indicator to a forced data character field or remove the statement and resubmit the job.

SORT058S, SUMMARIZATION FOR ALL INCLUDE SETS NOT THE SAME

Issued by: Sort/Merge

Explanation: This is a SORTRS job with multiple include sets. One of the following is true:

- Summary data fields were specified in the first include set, but not in all include sets.
- An overflow indicator (positions 1-3 contain FSV) was specified in the first include set, but not in all include sets.
- The number of summary fields in an include set is less than the number in the first include set.

System Action: Sort/Merge ends after the specification phase.

User Response: Determine if summarization is desired. If so, supply summary specifications for all include sets, consistent with the summary specifications for the first include set. Resubmit the job.

SORT059S, NUMBER OF CONTROL FIELDS FOR INCLUDE SET > 12

Issued by: Sort/Merge

Explanation: The accumulated number of control field specifications within the current include set exceeds 12.

System Action: Sort/Merge bypasses the referenced statement and continues the specification phase. The job ends after the specification phase.

User Response: Check the number of accumulated control fields for this include set. The number may not exceed 12. Correct the errors and resubmit the job.

SORT060S, CONTROL FIELD LENGTH FOR INCLUDE SET > HEADER VALUE

Issued by: Sort/Merge

Explanation: The accumulated length of control field statements within the current include set exceeds the length specified in positions 8-10 of the header statement.

System Action: Sort/Merge bypasses the referenced statement and continues the specification phase. The job ends after the specification phase.

User Response: Check the length specified in positions 8-10 of the header statement against the specified control fields. Correct the errors and resubmit job.

SORT061S, DATA LENGTH FOR SET NOT CONSISTENT WITH DS RECSIZE

Issued by: Sort/Merge

Explanation: This is a SORTR or SORTRS job. The data field specifications (positions 1 and 2 contain FD, FS, or FR) for this include set are not consistent with the record size defined in the output DSD statement. If you drop the control field (position 13 of the header statement contains a D), the sum of the lengths of the data fields cannot be greater than the size of the output record.

If you keep control fields (position 13 contains a K or is blank), the sum of the data field lengths plus the sum of the control field lengths (as specified in positions 8-10 of the header statement) cannot be greater than the size of the output record.

System Action: Sort/Merge continues processing and terminates the job after the specification phase.

User Response: Correct either (1) the record size defined in the output DSD statement or (2) the **from** and **to** fields (positions 4-11) in the field statement and resubmit the job.

**SORT062S, CONTROL FIELD DROPPED;
DATA FIELD NOT IN ALL INCLUDE
SETS**

Issued by: Sort/Merge

Explanation: This is a SORTRS job; for one or more include sets, no data to create an output record has been specified. One of the following conditions exist:

- Fixed input records are being processed, position 13 of the header statement specifies that control fields be dropped, and no data field statements (an FD, FR, or FS in positions 1-2 of a field statement) have been specified in one or more include sets.
- Variable records are being processed (for which control fields are always dropped) and no data field statements have been specified in one or more include sets.

System Action: The job ends after the specification phase.

User Response: If processing fixed records, either ensure that control fields are kept, or that data fields are specified for all include sets. If processing variable records, ensure that data fields are specified for all include sets.

**SORT063W, NO SUMMARY SPECS FOUND
IN FIRST INCLUDE SET - SORTRS JOB**

Issued by: Sort/Merge

Explanation: A SORTRS job (record summary sort) was specified on the header statement, however, no summary data field specifications were provided in the first include set for this job.

System Action: If the warning message option (position 14 of the header statement) is blank or T, the job ends after the specification phase. If it is P, the Sort/Merge operation completes and all records with duplicate control fields are eliminated from the output data set.

User Response: Check the job type to be sure that SORTRS was intended.

**SORT064S, ALTSEQ STATEMENT IN
WRONG ORDER; RECORD BYPASSED**

Issued by: Sort/Merge

Explanation: An ALTSEQ specification (A in position 1) is in the wrong order. It cannot follow an include statement, an omit statement or a field statement.

System Action: This statement is bypassed and processing continues with the next statement. The job ends after the specification phase.

User Response: Remove the statement from the specifications or place it in the proper order. Resubmit the job.

SORT065S, SPECIFICATION IN WRONG ORDER; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. If this is a control field statement (F in position 1), it improperly follows an omit statement (O in position 1). If this is an omit statement, it improperly follows an include statement (I in position 1).

System Action: This statement is bypassed and processing continues with next statement. The job ends after the specification phase.

User Response: Put the statement in the proper order and resubmit the job.

SORT066S, INCLUDE/OMIT SPEC AFTER INCLUDE-R; INCLUDE-R IGNORED

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The referenced statement is an include or omit statement (position 1 contains an I or O). An include-remaining statement has been previously read or has been implied (header or ALTSEQ followed by a control field statement). Only field statements (F in position 1) may follow include-remaining statements.

System Action: This statement and all following statements are processed as though the include-remaining statement had not been read. The job ends after the specification phase.

User Response: Change the include-remaining statement to an include statement or remove the referenced include/omit statement and all following statements. Resubmit the job.

SORT067T, LAST SET NOT A VALID INCLUDE SET

Issued by: Sort/Merge

Explanation: The last set of your specifications should have been an include set. However one of the following types of errors occurred:

1. There were no statements after the header statement
2. There were no statements after the DSD statements
3. There were no statements after the ALTSEQ statement
4. The last set was an omit set
5. The last set was an include set, but it contained no field statements

System Action: The job terminates immediately.

User Response: Make one of the following corrections depending upon the type of error that occurred:

1. Supply the statement(s) following the header statement
2. Supply the statement(s) following the DSD statements
3. Supply the statement(s) following the ALTSEQ statement
4. Eliminate this omit set as the records are omitted by default
5. Supply the missing field statements

Resubmit the job.

S

SORT068T, NO STORAGE AVAILABLE, TOO MANY SPECIFICATIONS

Issued by: Sort/Merge

Explanation: The main storage allocated by Sort/Merge to process the specification statements has been exceeded.

System Action: The job ends with no additional specification processing.

User Response: Either (1) increase the dynamic storage allotted to Sort/Merge or (2) decrease the number of specifications provided in the job and resubmit the job.

SORT069T, RECORD FORMAT INVALID, POSITIONS 18-19

Issued by: Sort/Merge

Explanation: The record format specified in positions 18-19 of the referenced DSD statement is not an F (followed by a blank), FB, V (followed by a blank), or VB.

System Action: Sort/Merge terminates the job.

User Response: Specify a F, FB, V or VB in positions 18-19 of the DSD statement and resubmit the job.

SORT070T, BLOCK SIZE INVALID, POSITIONS 25-29

Issued by: Sort/Merge

Explanation: The block size specified on the referenced DSD statement (positions 25-29) is invalid for one of the following reasons:

- The field contains a non-numeric character or embedded blanks
- The field entry is not right justified
- The field is blank
- The block size is greater than 32,767
- Positions 18-19 of the referenced statement are F or FB, and the block size specified is not greater than 0
- Positions 18-19 of the referenced statement are V or VB, and the block size specified is not greater than 8

System Action: Sort/Merge terminates the job.

User Response: Correct the block size and resubmit the job.

**SORT071T, RECSIZE/BLKSIZE
INCOMPATIBLE**

Issued by: Sort/Merge

Explanation: The record size (in positions 20-24) and the block size (in positions 25-29) specified on the referenced DSD statement are incompatible for one of the following reasons:

- Positions 18-19 are FB, and the block size is not a multiple of the record size
- Positions 18-19 is V, and the block size is not equal to the record size plus 4
- Positions 18-19 are VB and the block size minus 4 is not a multiple of the record size

System Action: Sort/Merge terminates the job.

User Response: Correct either the record size or the block size to ensure consistency and resubmit the job.

**SORT072T, RECORD SIZE INVALID,
POSITIONS 20-24**

Issued by: Sort/Merge

Explanation: The record size specified on the referenced DSD statement (in positions 20-24) is invalid for one of the following reasons:

- The field contains a non-numeric character or embedded blanks
- The field entry is not right justified
- The field is blank
- Positions 18-19 of the referenced statement are F or FB, and the record size specified is < 1 or $> 32,767$
- Positions 18-19 of the referenced statement are V or VB, and the record size specified is < 5 or $> 32,763$.

System Action: Sort/Merge terminates the job.

User Response: Correct the record size and resubmit the job.

S

SORT075P, SPECIFICATION PHASE ENDED

Issued by: Sort/Merge

Explanation: The specification phase of Sort/Merge has ended. Specification statements have been syntax checked.

System Action: If severe errors were found, the job ends after the specification phase. If warning errors were found, the job may or may not proceed to the execution phase, depending on the warning message option on the header specification. If no errors were detected, the job proceeds to the execution phase of Sort/Merge.

User Response: This is an informational message. No user response is needed.

SORT076P, INPUT PHASE STARTED

Issued by: Sort/Merge

Explanation: The execution phase of Sort/Merge has begun. Input records will be selected, work records built and written to the work data set.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT079T, MERGE INPUT DATA SET OUT OF SEQUENCE: - xxx

Issued by: Sort/Merge

Explanation: This is a MERGE job. All input data sets to be merged must be in sequence, either ascending or descending. The xxx specifies the data set and volume name that was out of sequence.

System Action: The job ends.

User Response: Ensure that each input data set to be merged is in sequence, and check the header statement to be sure that the correct sort sequence (A or D in position 11) was specified. Resubmit the job.

SORT082P, INPUT PHASE ENDED xxx yyy zzz

Issued by: Sort/Merge

Explanation: Input records have been read. The records to be sorted or merged have been selected and formatted.

The numbers represented by xxx, yyy, and zzz are aids for debugging an abnormal end of Sort/Merge processing.

At the end of execution segment 1:

- xxx is the number of work records created
- yyy is the number of sorted strings created
- zzz is the work block start number

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT083P, INTERMEDIATE MERGE
PHASE STARTED xxx yyy zzz**

Issued by: Sort/Merge

Explanation: Records are being read and rewritten to the work data set.

The numbers represented by xxx, yyy, and zzz are aids for debugging an abnormal end of Sort/Merge processing.

At the beginning of execution segment 2:

- xxx is the number of sorted strings created
- yyy is the calculated order of merge for the intermediate merge
- zzz is the calculated order of merge for the final merge.

The order of merge is the number of sorted strings, as calculated by the program, that will be brought into storage at any one time from the work data set.

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT084P, INTERMEDIATE MERGE
PHASE ENDED xxx**

Issued by: Sort/Merge

Explanation: The intermediate merge phase has completed successfully.

The number represented by xxx is an aid for debugging an abnormal end of Sort/Merge processing. At the end of execution segment 2, xxx sorted strings remained to be merged in the final merge pass (execution segment 3).

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT085P, FINAL MERGE PHASE
STARTED xxx yyy zzz****Issued by:** Sort/Merge**Explanation:** The final pass of this sort job has begun. Records are being written to your output file.

The numbers represented by xxx, yyy, and zzz are aids for debugging an abnormal end of Sort/Merge processing. At the start of execution segment 3:

- xxx is the number of strings to be merged
- yyy is the calculated order of merge for the intermediate merge
- zzz is the calculated order of merge for the final merge

The order of merge is the number of sorted strings, as calculated by the program, that will be brought into storage at any one time from the work data set.

System Action: None.**User Response:** This is an informational message. No user response is needed.**SORT086P, FINAL MERGE PHASE ENDED****Issued by:** Sort/Merge**Explanation:** The last pass of this sort was just successfully completed.**System Action:** None.**User Response:** This is an informational message. No user response is needed.**SORT088N, RECORDS READ FROM INPUT
DATA SET(S) xxx****Issued by:** Sort/Merge**Explanation:** xxx is the number of records that Sort/Merge read from the input data sets.**System Action:** None.**User Response:** This is an informational message. No user response is needed.**SORT089N, RECORDS INSERTED BY
INPUT EXIT ROUTINE: xxx****Issued by:** Sort/Merge**Explanation:** This is a SORTR or SORTRS job. xxx is the number of input records a user has passed to Sort/Merge via the input exit routine.**System Action:** None.**User Response:** This is an informational message. No user response is needed.**SORT090N, RECORDS DELETED BY INPUT
EXIT ROUTINE: xxx****Issued by:** Sort/Merge**Explanation:** xxx is the number of records the user has deleted from the input data set via the user input exit routine.**System Action:** None.**User Response:** This is an informational message. No user response is needed.

**SORT091N, RECORDS INSERTED BY
OUTPUT EXIT ROUTINE: xxx**

Issued by: Sort/Merge

Explanation: This is a SORTR, SORTRS or MERGE job. xxx is the number of records added or inserted to the sorted output by the output exit routine.

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT092N, RECORDS DELETED BY
OUTPUT EXIT ROUTINE: xxx**

Issued by: Sort/Merge

Explanation: xxx is the number of records deleted from the sorted output by the output exit routine.

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT093N, RECORDS WRITTEN TO
OUTPUT DATA SET: xxx**

Issued by: Sort/Merge

Explanation: xxx is the number of records which Sort/Merge has written to the output data set.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT094N, I/O ERRORS ACCEPTED: xxx

Issued by: Sort/Merge

Explanation: One of two conditions exist:

- Sort/Merge was initiated from a user program with an I/O exit routine. The routine has passed an action code after returning control to Sort/Merge indicating that Sort/Merge is to accept all I/O errors.
- You have specified an A in position 17 of the header statement indicating that you want Sort/Merge to accept all I/O errors.

xxx I/O errors have occurred after Sort/Merge issued one or more READ or WRITE, or both instructions.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT095N, I/O ERRORS SKIPPED: xxx

Issued by: Sort/Merge

Explanation: An I/O error has occurred during a read or write operation. You specified either on the header specification or in a user exit routine, that the I/O error will be skipped (the buffer contents are not used). The value xxx is the number of I/O errors skipped.

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT097P, NORMAL ENDING FOR
SORT/MERGE PROCESSING**

Issued by: Sort/Merge

Explanation: Sort/Merge has completed successfully.

System Action: None.

User Response: This is an informational message. No user response is needed.

**SORT098S, NUMBER OF CONDITIONALS
IN MULTI-CONDITIONAL TEST > 255**

Issued by: Sort/Merge

Explanation: The referenced statement is a field statement specifying a conditional force. It has exceeded the maximum number of conditional force-field statements allowed in a multi-conditional force. The maximum number of conditional force-field statements in a multi-conditional test is 255.

System Action: Sort/Merge continues processing. The job ends after the specification phase.

User Response: Limit the number of conditionals in the multi-conditional test to 255. Separate the current multi-conditional test into two or more tests, and resubmit the job.

SORT099P,----+----1----+----2

Issued by: Sort/Merge

Explanation: This message is a positioning line which corresponds to the positions on a specification statement. If the print option on the header statement (position 12) contains a O or is blank, this line is printed immediately before the header statement. It can be used as a guide in determining if the entries in the specification statements are in the correct position.

System Action: None

User Response: This is an informational message. No user response is needed.

**SORT101T, RETURN CODE ISSUED: xxx
yyy**

Issued by: Sort/Merge

Explanation: After issuing the instruction xxx, Sort/Merge received a return code of yyy.

System Action: The job terminates.

User Response: Read the explanation of the xxx instruction return code. If a user error occurred, correct it and resubmit the job.

**SORT102P, RETURN CODE ISSUED: xxx
yyy zzz**

Issued by: Sort/Merge

Explanation: After issuing the I/O instruction xxx, Sort/Merge received a return code of yyy. The data set being processed was zzz.

System Action: If provided, the user I/O error exit routine determines disposition of the error. Otherwise, Sort/Merge takes the option specified in position 17 of the header statement.

User Response: Read the explanation of the xxx instruction return code. If a user error occurred, correct it and resubmit the job.

**SORT104T, SORT/MERGE TERMINATED
DUE TO I/O ERROR**

Issued by: Sort/Merge

Explanation: One of three conditions exists:

- An I/O error has occurred and you have instructed Sort/Merge, either through an I/O error exit routine or in position 17 of the header statement, to terminate processing.
- An I/O error has occurred 10 consecutive times and Sort/Merge has terminated regardless of the options requested by the I/O exit routine or by the header statement.
- An I/O error has occurred while processing the work data set.

System Action: The job terminates.

User Response: Reallocate the data set, if possible. If you have specified *terminate*, you might want to change your option and accept the errors.

**SORT105T, NOT ENOUGH STORAGE TO
BUILD BUFFERS**

Issued by: Sort/Merge

Explanation: There is insufficient main storage to build buffers in one or more of the Sort/Merge execution phases.

System Action: The job terminates.

User Response: Increase the amount of dynamic storage available to Sort/Merge for execution, either through the LOAD instruction, a \$JOBUTIL statement, or the SS command of \$DISKUT2, depending on how Sort/Merge was invoked. See the *Language Reference* or *Operator Commands and Utilities Reference* for details.

Other ways of getting more storage for buffers is to:

- Decrease the blocking factor of the output data set.
- Reblock the input data set using a smaller block size.
- Try to decrease the control field lengths or the output record length.

**SORT112S, DEVICE TYPE INVALID,
POSITION 3; DISK/DISKETTE
ASSUMED**

Issued by: Sort/Merge

Explanation: The device type specified in position 3 of the referenced DSD statement is incorrect. Check for one of the following:

- Not blank, D, or T
- A disk or diskette is not specified for the work file
- Tape is specified for input and the header statement (column 15) indicates the input is indexed

System Action: Sort/Merge assumes a device type of disk or diskette and continues processing. The job ends after the specification phase.

User Response: Enter the correct character (blank, D, or T), ensure compatibility with the header statement, and resubmit the job.

**SORT113T, INVALID ACTION CODE FROM
USER I/O EXIT ROUTINE**

Issued by: Sort/Merge

Explanation: The action code issued with the RETURN instruction from a user I/O error exit routine is not a 0 (for terminate), a 4 (for accept), or an 8 (for skip).

System Action: Sort/Merge terminates the job.

User Response: Ensure that your I/O error exit routine provides the correct action code.

**SORT116T, INVALID DATA IN CONTROL
FIELD FOR SPECIFIED DATA TYPE**

Issued by: Sort/Merge

Explanation: The data type (packed decimal or unpacked decimal) in position 2 of the field statement does not match the data actually found in the control field.

System Action: Sort/Merge terminates the job.

User Response: Correct either (1) the data type specification (position 2) or (2) the field location (positions 4-11 in the field statement), or both, and resubmit the job.

**SORT117T, FIELD NOT WITHIN INPUT
RECORD BOUNDARIES**

Issued by: Sort/Merge

Explanation: A variable length record is being processed. Either a data field or a control field specified in a field statement does not lie within the input record.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the position locations you specify are within the length of the input record and resubmit the job.

**SORT118W, INVALID DATA FOR
SUMMARY DATA FIELD**

Issued by: Sort/Merge

Explanation: The data type found in the summary data field does not agree with the type of data specified in position 3 of the summary data field statement; either the sign portion, or the data portion, or both are incorrect. Sort/Merge issues this message only once, regardless of the number of times it finds invalid data.

System Action: Sort/Merge summarizes the field, but the results will be unpredictable, and continues processing.

User Response: Ensure that you have specified in the field statement (1) the correct data type in position 3 and (2) the correct **to** and **from** positions (as specified in positions 4-11 of the field statement) that define the location of the summary field in the input record. Check also if the data in the input record is valid.

**SORT120W, INVALID ACTION CODE FROM
USER OUTPUT EXIT**

Issued by: Sort/Merge

Explanation: Sort/Merge was initiated from a user program with the LOAD instruction; bytes 4-5 of the user parameter list in the program contains the address of a user output exit routine. Sort/Merge has passed control to the output exit routine. Upon returning control to Sort/Merge the exit routine did not provide a valid action code.

System Action: Sort/Merge writes the records already processed to the output data set you have specified. If the output exit routine is responsible for writing the sorted or merged output records, Sort/Merge terminates the job.

User Response: Ensure that an appropriate action code of 0, 1, or 3 is returned to Sort/Merge from the output exit and resubmit the job.

**SORT121W, INSERT ACTION CODE
INVALID IN SORTA JOB**

Issued by: Sort/Merge

Explanation: This is a SORTA (record address sort) job. Either a user input exit routine or an output exit routine has provided an invalid action code.

Note: SORTA does not permit inserting records, thus, if you have specified an address in the return information, you will receive this message.

System Action: Sort/Merge continues processing without returning to the user exit.

User Response: If necessary, correct the code in your exit routine and resubmit the job.

**SORT122W, INSERT FAILED FOR
VARIABLE OUTPUT RECORD - TOO
LARGE**

Issued by: Sort/Merge

Explanation: Sort/Merge was initiated from a user program with the LOAD instructions; bytes 4-5 of the user parameter list indicate the address of an output exit routine.

System Action: Your output exit routines attempted to pass a variable-length record to Sort/Merge. The sector descriptor of the records shows a record length greater than the maximum length specified in the definition for the output data set.

If you have specified an output data set, Sort/Merge writes the records already processed to the output data set. If you have not specified an output data set, but instead your output exit routine is responsible for output, Sort/Merge terminates processing.

User Response: Ensure that the length of the variable output records and the maximum length established in the definition for the output data set conform, and then resubmit your job.

**SORT123W, ILLOGICAL ACTION CODE
FROM OUTPUT EXIT**

Issued by: Sort/Merge

Explanation: The action code returned by the output exit routine indicates that Sort/Merge should take an action that is either not possible, or not logical, or both. For example, on passing control to the exit routine, Sort/Merge may have indicated that all the output records have been processed, and your routine may have indicated in the action code that Sort/Merge is to process the record just passed.

System Action: If you have specified an output data set, Sort/Merge writes the records already processed to the output data set.

User Response: Check the logic of your routine in providing the action code. Correct the program and resubmit the job.

**SORT124W, INVALID ACTION CODE FROM
USER INPUT EXIT**

Issued by: Sort/Merge

Explanation: Your input exit routine passed an invalid action code to Sort/Merge.

System Action: Sort/Merge continues processing the records already read from the input data set and does not return control to your input exit routine.

User Response: Check for the valid input exit action codes.

**SORT125W, INSERT FAILED FOR
VARIABLE INPUT RECORD - TOO
LARGE**

Issued by: Sort/Merge

Explanation: Sort/Merge was initiated from a user program with the `CALL` or `LOAD` instruction; bytes 4-5 of the user parameter list indicate the address of a user exit routine. An input exit routine is attempting to pass a variable-length record to Sort/Merge. The segment descriptor of the record shows a record length greater than the maximum length specified for the input data set or in the DSD list.

System Action: Sort/Merge continues processing the records already read and does not return control to the input exit routine.

User Response: Ensure that the length of the variable records passed from the user exit routine do not exceed the maximum length specified.

**SORT126W, ILLOGICAL ACTION CODE
FOR USER INPUT EXIT**

Issued by: Sort/Merge

Explanation: The return code from an input exit routine returned to Sort/Merge should take an action that is either not possible, or not logical, or both. For example, on passing control to your exit, Sort/Merge may have indicated that all the records in the input data set have been read in, and you may have indicated in the input exit action code that Sort/Merge is to process the record just read.

System Action: Sort/Merge processes the records already read from the input data set and does not return control to your input exit routine.

User Response: Check the logic of your program in providing the input exit action code. Correct the program and resubmit the job.

**SORT127T, RETURN CODE WAS ISSUED
FROM INSTRUCTION FOR SPEC PHASE
xxx yyy**

Issued by: Sort/Merge

Explanation: An I/O error occurred while processing the specification data set. After issuing the instruction `yyy`, Sort/Merge received a return code of `xxx`.

System Action: The job terminates.

User Response: Read the explanation of the `yyy` instruction return code. Correct and resubmit the job.

S

SORT128P, TABLE OVERFLOW CONDITION BEING PROCESSED

Issued by: Sort/Merge

Explanation: This is a SORT job. During the intermediate merge phase, a table internal to sort processing has become filled. Sort/Merge performs remedial processing to allow processing to continue.

System Action: The job runs to completion normally.

User Response: This is an informational message. No user response is needed.

SORT129T, ATTRIBUTES NOT SAME FOR ALL INPUT DATA SETS

Issued by: Sort/Merge

Explanation: This is a MERGE job or a sort job with more than one input data set. The record format or the record size specified in the DSD statements are inconsistent. The record formats must fall into one of two categories: (1) fixed (F) or fixed block (FB) or (2) variable (V) or variable block (VB). The record length must be the same for all the input data sets.

System Action: Sort/Merge terminates the job.

User Response: Ensure the input data sets meet the above restrictions and then resubmit the job.

SORT130S, NUMBER OF OUTPUT OR WORK DSD SPECIFICATIONS > 1; RECORD BYPASSED

Issued by: Sort/Merge

Explanation: Only one work DSD and one output DSD statement are valid. A work or output DSD statement has been previously specified.

System Action: Sort/Merge bypasses the referenced DSD statement and continues processing. The job ends after the specification phase.

User Response: Ensure that only one work DSD and only one output DSD is defined. Resubmit the job.

SORT131T, OUTPUT DATA SET FORMAT INCOMPATIBLE WITH INPUT

Issued by: Sort/Merge

Explanation: The record format defined for the output data set does not conform to the record format of the input data set(s).

System Action: Sort/Merge terminates the job.

User Response: Ensure that the record formats of the input and output data sets are both either variable or fixed, and resubmit the job.

**SORT132S, SORTA - MORE THAN ONE
INPUT DSD PROVIDED; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: This is a record address sort (SORTA), for which only one input data set is allowed. More than one input data set has been specified for this job.

System Action: Sort/Merge bypasses the referenced statement and continues processing. The job ends after the specification phase.

User Response: Provide only one input DSD statement (a DI in positions 1-2) and resubmit the job.

**SORT133T, NO WORK DATA SET
PROVIDED - SORT JOB**

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR, or SORTRS job. No work data set has been defined.

System Action: Sort/Merge terminates the job.

User Response: Supply a DSD statement for the work data and resubmit the job.

**SORT134S, NUMBER OF INPUT DSD
SPECIFICATIONS > 8; RECORD
BYPASSED**

Issued by: Sort/Merge

Explanation: This is a SORTR, SORTRS or MERGE job, for which a maximum of 8 input data sets is allowed. More than 8 input data sets have been defined for this job.

System Action: Sort/Merge bypasses the referenced statement and continues processing. The job ends after the specification phase.

User Response: Ensure that only 1 to 8 input data sets are defined and resubmit the job.

**SORT135T, FEWER THAN 2 INPUT DATA
SETS SPECIFIED FOR MERGE JOB**

Issued by: Sort/Merge

Explanation: This is a MERGE job, which requires from 2 to 8 input data sets. Less than two data sets have been specified.

System Action: Sort/Merge terminates the job.

User Response: Provide an additional input DSD statement (a DI in positions 1-2) and resubmit the job.

SORT136T, SORTR/SORTRS: NO INPUT DATA SET OR USER EXIT SPECIFIED

Issued by: Sort/Merge

Explanation: This is a SORTR or SORTRS job and the program has no input records to process because of one of the following conditions:

- If Sort/Merge was initiated with the \$L operator command or the \$JOBUTIL utility, no input DSD specifications were provided.
- If Sort/Merge was initiated from a user program with the LOAD instruction, then bytes 2-3 of the user parameter list in the program do not specify the address of an input exit routine, and no input DSD specifications were provided.

System Action: Sort/Merge terminates the job.

User Response: Either supply a user input exit routine through which you supply records to Sort/Merge or supply input DSD specifications.

SORT137T, RECORD FORMAT/LENGTH SPECIFICATION IN PARM LIST NOT VALID

Issued by: Sort/Merge

Explanation: Sort/Merge was initiated from a user program with the LOAD instruction. One of the following error conditions exists in the parameter list:

- Byte 8 does not contain a 0 (indicating a fixed record format) or a 1 (indicating a variable record format).
- Bytes 10-11 contain a 0, indicating a record length of zero.
- Byte 8 does not contain a 0 (indicating a fixed record format) and the header statement indicates that indexed data sets are being processed.

System Action: Sort/Merge terminates the job.

User Response: Correct the parameter list and resubmit the job.

SORT138T, NO OUTPUT DATA SET OR USER EXIT SPECIFIED

Issued by: Sort/Merge

Explanation: The program has no data set for the output records, due to one of the following conditions:

- If Sort/Merge was initiated with the \$L operator command or the \$JOBUTIL utility commands, no output DSD specification was provided.
- If Sort/Merge was initiated from a user program with the LOAD instruction, then bytes 4-5 of the user parameter list in the program do not specify the address of an output exit routine and no output DSD specification was provided.

System Action: Sort/Merge terminates the job.

User Response: Either supply a user output exit routine through which you will write the Sort/Merge output records or supply an output DSD specification.

SORT139T, THE OUTPUT DATA SET WILL OVERWRITE AN INPUT DATA SET

Issued by: Sort/Merge

Explanation: This is a SORTA or MERGE job. The Sort/Merge output data set will overwrite the Sort/Merge input data set of the same name, preventing access to the input records by the program.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the output data set and input data set(s) are not the same. Resubmit the job.

SORT140T, SORTA-NO INPUT DATA SET PROVIDED

Issued by: Sort/Merge

Explanation: This is a SORTA job, for which one input data must be provided. No input data set was specified.

System Action: Sort/Merge terminates the job

User Response: Supply a DSD statement for the input data set and resubmit the job.

SORT141T, THE WORK DATA SET WILL OVERWRITE AN INPUT DATA SET

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The Sort/Merge work data set will overwrite a Sort/Merge input data set of the same name, preventing access to the input records by the program.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the work data set and the input data set(s) are not the same. Resubmit the job.

S

SORT142T, INDEXED ORGANIZATION IS NOT SUPPORTED

Issued by: Sort/Merge

Explanation: This is a SORTR, SORTRS, or MERGE job. The data set organization in the header statement indicates that Indexed Access Method data sets are being supplied to Sort/Merge, but Sort/Merge was installed without Indexed Access Method support.

System Action: Sort/Merge terminates the job.

User Response: Ensure that Release 1.1 or later of the Sort/Merge Licensed Program, which contains support for the Indexed Access Method, is installed according to the instructions in the program directory.

SORT143T, OUT OF SPACE ON WORK DATA SET xxx

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR, or SORTRS job. The amount of space allocated to the work data set xxx is not sufficient to hold all the work records being processed.

System Action: Sort/Merge terminates processing without completing the sort and without writing any records to the output data set.

User Response: Recreate the work data set, increasing the allocation of space. See the *Sort/Merge Programmer's Guide* for guidelines in determining the amount of space required.

SORT144T, OUT OF SPACE ON OUTPUT DATA SET xxx

Issued by: Sort/Merge

Explanation: The physical end of the output data set xxx has been reached and there are more output records to be written.

System Action: The job terminates.

User Response: Increase the space allocated to the output data and resubmit the job.

SORT145W, INPUT DATA SET CONTAINS NO DATA xxx

Issued by: Sort/Merge

Explanation: This is a MERGE job. Sort/Merge found no data in the data set xxx, which was specified to contain the input data to be sorted or merged for this job.

System Action: Sort/Merge continues processing.

User Response: If the input data set should have contained data, take corrective action and resubmit the job.

SORT146T, INSUFFICIENT STORAGE TO INITIATE SPECIFICATION PHASE

Issued by: Sort/Merge

Explanation: Insufficient dynamic storage exists for the specification phase to execute.

System Action: Sort/Merge terminates processing.

User Response: Increase the amount of dynamic storage available to Sort/Merge for execution either through the LOAD instruction, a \$JOBUTIL statement, or the SS command of \$DISKUT2, depending on how Sort/Merge was invoked. See the *Language Reference* or *Operator Commands and Utilities Reference* for details.

SORT147T, THE OUTPUT DATA SET WILL OVERWRITE THE WORK DATA SET

Issued by: Sort/Merge

Explanation: This is a SORTA, SORTR or SORTRS job. The Sort/Merge output data set will overwrite the Sort/Merge work data set, preventing access to the work records.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the work data set and the output data are not the same. Resubmit the job.

SORT149N, RECORDS SORTED OR MERGED: xxx

Issued by: Sort/Merge

Explanation: xxx records have been selected (as determined by include/omit statements or an exit routine, when present) from the input data sets for Sort/Merge processing.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT150P, PROCESSING TERMINATED; ERROR CONDITION ENCOUNTERED

Issued by: Sort/Merge

Explanation: Sort/Merge has terminated the job after encountering one or more of the following error conditions:

- A terminal error during the specifications or execution phase.
- A severe error during the specification phase.
- A warning message during the specification phase; you have specified a T in position 14 of the header statement indicating you want Sort/Merge to stop processing at the end of the specification phase if any warning messages have been issued.

System Action: None.

User Response: Check the errors indicated by the messages, correct them, and resubmit the job.

SORT151P, MERGE PROCESSING ENDED

Issued by: Sort/Merge

Explanation: This is a MERGE job. Sort/Merge has completed processing the input records.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT152P, MERGE PROCESSING STARTED

Issued by: Sort/Merge

Explanation: This is a MERGE job. Sort/Merge has started processing the input records.

System Action: None.

User Response: This is an informational message. No user response is needed.

SORT154T, INSUFFICIENT STORAGE TO INITIATE EXECUTION PHASE

Issued by: Sort/Merge

Explanation: Insufficient dynamic storage exists for the execution phase.

System Action: Sort/Merge terminates processing.

User Response: Increase the amount of dynamic storage available to Sort/Merge for execution either through the LOAD instruction or a \$JOBUTIL statement, depending on how Sort/Merge was invoked. Refer to the *Language Reference* or the *Operator Commands and Utilities Reference* manual for details.

SORT160T, AN I/O ERROR OCCURRED DURING OPEN xxx yyy

Issued by: Sort/Merge

Explanation: An I/O error occurred while trying to open the data set xxx located on volume yyy.

System Action: Sort/Merge terminates the job.

User Response: Correct the device problem or change the associated DSD specifications statement.

SORT161T, I/O ERROR HAS OCCURRED DURING SETEOD xxx yyy

Issued by: Sort/Merge

Explanation: An I/O error occurred while trying to set the end-of-data for the output data set xxx located on volume yyy.

System Action: Sort/Merge terminates the job.

User Response: Check the data set on which the I/O error occurred for possible errors and correct them.

SORT162T, OPEN FAILED; DATA SET NOT FOUND xxx yyy

Issued by: Sort/Merge

Explanation: The data set xxx could not be found on volume yyy, as specified in the DSD statement.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the data set exists; if necessary, correct the DSD statement.

**SORT163T, OPEN FAILED; DEVICE
ADDRESS INCORRECT xxx yyy**

Issued by: Sort/Merge

Explanation: The address of the device containing the data set xxx located on volume yyy is invalid.

System Action: Sort/Merge terminates the job.

User Response: Check the data set on which the error occurred for possible errors and correct them.

**SORT164T, OPEN FAILED; VOLUME
INVALID xxx yyy**

Issued by: Sort/Merge

Explanation: The volume yyy specified in the DSD statement could not be found. The volume yyy specified for the data set xxx in a DSD statement could not be found.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the volume specified exists; if necessary correct the DSD statement.

**SORT165T, OPEN FAILED; LIBRARY NOT
FOUND xxx yyy**

Issued by: Sort/Merge

Explanation: The library for the data set xxx and volume yyy could not be found.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the data set and volume exists; correct the DSD statement if necessary.

**SORT166T, OPEN FAILED; VTOC IN ERROR
xxx yyy**

Issued by: Sort/Merge

Explanation: While trying to open the data set xxx on volume yyy, the VTOC for the device could not be read; it may have been destroyed.

System Action: Sort/Merge terminates the job.

User Response: Ensure that the DSD statement is correct and the data set and volume exist as specified. If the problem recurs, there could be a device problem.

**SORT167T, SORT TERMINATED BY USER
REQUEST**

Issued by: Sort/Merge

Explanation: A user exit routine has returned to Sort/Merge an action code which instructs Sort/Merge to terminate.

System Action: Sort/Merge terminates the job.

User Response: This is an informational message. No user response is needed.

S

SOURCE AND TARGET CANNOT BE THE SAME

Issued by: \$COPY

Explanation: You have specified the same data set name for the source and target, on the same volume. The data set names can be the same as long as they are on different volumes.

System Action: Ends the operation.

User Response: Specify two different data set names, or use two different volumes for source and target.

SOURCE AND TARGET DATA SETS DO NOT HAVE THE SAME ORGANIZATION

Issued by: \$COPYUT1, \$COPY

Explanation: Both the source and target data set must be of the same type: either data type or program type, as specified when the data sets were allocated.

System Action: Ends the operation.

User Response: Retry the function using two different data sets of the same type, or reallocate one of the original data sets, changing its organization to match the other.

SOURCE AND TARGET VOLUME SIZES ARE NOT EQUAL

Issued by: \$MOVEVOL

Explanation: The sizes of both volumes must be equal for the requested function to take place.

System Action: Ends the operation.

User Response: Retry the function using two different volumes that match in size. Or, using the original volumes, reallocate one so that its size matches the other.

SOURCE DATA SET HAS xxx RECORDS TARGET VOLUME HAS ONLY yyy RECORDS

Issued by: \$COPY

Explanation: The size of the target volume is too small to contain a copy of the source data set.

System Action: Returns to command mode.

User Response: Reallocate the target volume to accommodate the size of the source data set, or use a different, larger volume.

**SOURCE VOLUME HAS xxx RECORDS
TARGET DATA SET HAS ONLY yyy
RECORDS**

Issued by: \$COPY

Explanation: The size of the target data set is too small to hold a copy of the source volume.

System Action: Returns to command mode.

User Response: Reallocate the target data set so its size can accommodate the source volume, or use a different, larger data set.

**SPACE FOR THE SUPERVISOR IS TOO
SMALL YOU MUST REORGANIZE
LIBRARY**

Issued by: \$UPDATEH

Explanation: Not enough space was allocated to the supervisor.

System Action: Returns to command mode.

User Response: Allocate more space for the supervisor and retry the function.

**SPECIFIED COPYCODE MODULE IS NOT A
DATA SET**

Issued by: \$EDXASM

Explanation: The copy code module was found but the data set type is not "data". All copy code data sets must have a type of "data" (as opposed to "program" type).

System Action: The error is flagged and compilation continues.

User Response: Ensure the copy code data set specified is a data set with a type of "data".

**SPECIFY WAIT=YES OR WAIT=NO FOR
DISK OPERATIONS**

Issued by: \$EDXASM

Explanation: The WAIT= operand on the READ/WRITE instruction was specified incorrectly.

System Action: The error is flagged and compilation continues.

User Response: Specify either WAIT=YES (suspend task until operation is complete) or WAIT=NO (control is returned after operation starts). If omitted, WAIT=YES is the default.

S

SPECPI= MUST BE SPECIFIED FOR TYPE=GROUP AND TYPE=BIT

Issued by: \$EDXASM

Explanation: If TYPE=GROUP or TYPE=BIT is specified for a process interrupt (PI) IODEF, the special process interrupt routine operand (SPECPI) must be present.

System Action: The error is flagged and compilation continues.

User Response: Specify the label of the special process interrupt routine on the SPECPI= operand.

SPOOL ACTIVE LIMIT REACHED

Issued by: Spooling

Explanation: An application program enqueued a spool device, which allocated the last free spool control block available. A free spooling control block becomes available when a program with an active spool job terminates, or when a spool job goes from active to ready status.

System Action: The next program that enqueues on a spool device waits for a free control block to process the spool job.

User Response: To prevent this situation from recurring in future spool sessions, run the \$SPLUT1 utility program, increasing the maximum number of allowable active spool jobs.

SPOOL DATA SET FULL

Issued by: Spooling

Explanation: Request for spool data set space on behalf of a task is unsuccessful because all of the spool data set records are allocated to existing spool jobs.

System Action: The requesting task waits for deallocation of a spool data set group by spooling.

User Response: To prevent this situation from recurring in future spool sessions, allocate a larger spool data set. The immediate alternatives are:

- Allow the program to continue and wait until space in the spool data set becomes available
- Start a spool writer to print a spool job (and eventually delete it from the spool data set)
- Delete a ready or printed spool job
- Release the held disposition (via \$\$ REL) from a held spool job allowing a spool writer to print it (and eventually delete it from the spool data set)
- Remove the keep disposition (via \$\$ KEEP N) from a kept spool job, allowing it to be deleted after being processed by a spool writer

SPOOL DATA SET HAS INVALID CHECKPOINT DATA, RESTART ABORTED

Issued by: Spooling

Explanation: When the spool facility tried to restart using data saved from a previous session, it found that the data was unusable.

System Action: The spool operation ends.

User Response: You can:

- Start spooling again from the beginning and rerun the spool job.
- Restart spooling with another spool data set.

SPOOL DS READ ERROR, RC=xxx

Issued by: Spooling

Explanation: An error occurred while reading from the spool data set. The result was a return code of xxx.

System Action: Returns to the program currently executing.

User Response: Check the disk and diskette read/write return codes to find the cause of the problem.

SPOOL DS WRITE ERROR, RC=50

Issued by: Spooling

Explanation: A spool data set group is required for a spool job, and all spool data set groups have been allocated. The message "SPOOL GROUP CAPACITY REACHED" will have been issued prior to this message.

System Action: Waits for space on the spool data set.

User Response: See the explanation for the "SPOOL GROUP CAPACITY REACHED" message and take the appropriate action.

SPOOL DS WRITE ERROR, RC=xxx

Issued by: Spooling

Explanation: An error occurred while writing to the spool data set. The result was a return code of xxx (any code except 50).

System Action: Returns to the program currently executing.

User Response: Check the spool return codes or the disk and diskette read/write return codes to find the cause of the problem.

SPOOL ERROR, RC=xxx

Issued by: Spooling

Explanation: A condition was detected during spooling initialization which prevents the session from continuing. The result was a spool return code of xxx.

System Action: The spool session ends and \$SPOOL is unloaded.

User Response: Check the spool return code and take the appropriate action.

- Release the held disposition (via \$\$ REL) from a held spool job allowing a spool writer to print it (and eventually delete it from the spool data set)
- Remove the keep disposition (via \$\$ KEEP N) from a kept spool job, allowing it to be deleted after being processed by a spool writer

SPOOL GROUP CAPACITY REACHED

Issued by: Spooling

Explanation: This is a warning that the last group of records in the spool data set was allocated to a spool job. Future requirements for a data set group cause the issuing program to wait for a data set group to be deallocated. A spool data set group is deallocated when a spool job is deleted from the system.

System Action: The program that issued the print instruction continues. Future print operations that result in a data set group allocation cause the issuing program to be placed in the wait state until a data set group becomes available.

User Response: To prevent this situation from recurring in future spool sessions, allocate a larger spool data set. The immediate alternatives are to:

- Allow the program to continue and wait until space in the spool data set becomes available
- Start a spool writer to print a spool job (and eventually delete it from the spool data set)
- Delete a ready or printing spool job

SPOOL INITIALIZATION COMPLETE

Issued by: Spooling

Explanation: Spooling initialization has been completed. Output may now be spooled.

System Action: Starts designated autostart spool writers, if any.

User Response: Enter \$\$ spooling commands, if necessary, after automatic starting of spool writers.

SPOOL JOB CAPACITY REACHED

Issued by: Spooling

Explanation: An application program enqueued a spool device which used the last free entry in the spool job queue. A free entry in the spool job queue becomes available when a spool job is deleted from the system.

System Action: The program that enqueued the spool device continues. Future enqueue operations to a spool device cause the issuing program to be placed in the wait state until a free entry in the spool job queue becomes available.

User Response: To prevent this situation from recurring in future spool sessions, run the \$SPLUT1 utility program, increasing the total maximum number of allowable spool jobs.

The immediate alternatives are to:

- Wait until a spool job queue entry becomes available
- Start a spool writer to begin printing (and eventually deleting) a spool job
- Delete a ready spool job
- Release the held disposition (via \$\$ REL) from a held spool job, allowing a spool writer to begin printing it (and eventually to delete it from the system)
- Remove the keep disposition (via \$\$ KEEP N) from a kept spool job, allowing it to be deleted

**SPOOL WRITER xxx TERMINAL I/O
ERROR RC=yyy ENTER Y TO RETRY
OR N TO TERMINATE WRITER:**

Issued by: Spooling

Explanation: The spool writer xxx encountered a terminal I/O error on its output device (the device name is always the same as that of the spool writer). The result was a terminal I/O general return code of yyy.

System Action: Waits for you to respond “Y” or “N”:

- If you respond “Y”, spool processing continues.
- If you respond “N”, the system ends the spool job being printed.

User Response: Check the terminal I/O general return code and correct the problem if possible. If the system ends the current spool print job because the problem was uncorrectable, you can redirect the job to another spool writer using the \$\$ ALT command.

START= OPERAND MUST BE SPECIFIED

Issued by: \$EDXASM

Explanation: The “start” operand must be specified on the instruction.

System Action: The error is flagged and compilation continues.

User Response: Specify for “start” the label where program execution is to begin.

**STATIC SCREENS ARE NOT SUPPORTED
FOR THIS TERMINAL TYPE**

Issued by: \$EDXASM

Explanation: The operand SCREEN=STATIC is not valid for this terminal type.

System Action: The error is flagged and compilation continues.

User Response: Refer to the TERMINAL statement in the *Installation and System Generation Guide* to determine which operands are valid for this device.

**STATION xxx, TERMINAL yyy NOT FOUND
IN TERMINAL FILE**

Issued by: \$MTM

Explanation: Either the terminal was not specified in the TERMINAL file or the terminal was specified incorrectly in the TERMINAL file.

System Action: Does not start the terminal as a Communications Facility station.

User Response: Ensure the terminal name is defined correctly in the TERMINAL file.

S

STATION xxx, TERMINAL yyy START ERROR RC=zzz

Issued by: \$MTM

Explanation: CDMINIT2 was unable to start a station and the station was not already started. The result was a return code of zzz.

System Action: The station identified as xxx will not be available through the Communications Facility.

User Response: The return code returned is from the Communications Facility command processor. Check the command processor messages in the Communications Facility operator guide to determine the problem and take the appropriate action.

STATION xxx, TERMINAL yyy STATION STARTED LOAD ERROR, RC=zzz STATION HALTED

Issued by: \$MTM

Explanation: The initialization routine (CDMINIT2) was unable to load an asynchronous task for the station it just started or found already active. The result was a LOAD return code of zzz.

System Action: The station just started will be halted. The storage message queue will be purged.

User Response: Check the LOAD return code to determine the problem and take the appropriate action.

STATIONS FILE DISK ERROR RC=xxx

Issued by: \$MTM

Explanation: A disk read error occurred while reading the STATIONS file. The result was a return code of xxx.

System Action: The initialization routine (CDMINIT2) is terminated and no Communications Facility stations are started.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

STORAGE SIZE MUST BE SPECIFIED (16K - 1024K)

Issued by: \$EDXASM

Explanation: The STORAGE= operand on the SYSTEM statement is required and must be specified within the range of 16K–512K bytes.

System Action: The error is flagged and compilation continues.

User Response: Code the STORAGE= operand and specify the storage size within 16K–512K bytes.

SUB1 INDEX PARAMETER ERROR

Issued by: \$BSCUT2

Explanation: The system detected an internal error.

System Action: Ends the operation.

User Response: Examine the output to find the problem, correct it and retry.

SUB2 PARAMETER ERROR

Issued by: \$BSCUT2

Explanation: The system detected an internal error.

System Action: Ends the operation.

User Response: Examine the output to find the problem, correct it and retry.

SUBMIT ABORTED - FILE ERROR xxx

Issued by: \$RJESNA

Explanation: A disk error occurred while a job stream was being transmitted to the host, or an I/O error occurred during the opening of the data set. The result was a return code of xxx.

System Action: The system cancels the job if it was submitted or does not submit the job.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

SUPERVISOR AND COMMON AREA IN PARTITION # xxx EXCEEDS MAXIMUM SIZE OF 64K BY yyy (zzz HEX)

Issued by: \$XPSLINK from \$XPSPOST

Explanation: The amount of common area and the size of the supervisor assigned together are too large to be contained in one partition.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Either decrease the size of the common area or move part of the supervisor to another partition.

SUPPORT DOES NOT EXIST FOR xxx

Issued by: IOLOADER

Explanation: A SENSORIO configuration statement could not be found for the Sensor I/O device xxx defined in the IODEF statement.

System Action: Program load terminates.

User Response: Correct the IODEF or SENSORIO statement and retry.

SUPPORT DOES NOT EXIST FOR xxx

Issued by: \$IOTEST

Explanation: The address you specified does not support any device in your system.

System Action: Ends the operation.

User Response: Specify a valid device address. To obtain valid addresses, use the LD command of \$IOTEST.

SYMBOL EXCEEDS 8 CHARACTERS IN LENGTH

Issued by: \$EDXASM

Explanation: A symbol used in the program exceeded the maximum length. Valid symbols must be 1–8 characters in length.

System Action: The error is flagged and compilation continues.

User Response: Specify the symbol as 1–8 characters in length.

SYSTEM VOLUME NAME...ACCESS DENIED

Issued by: \$COMPRES,\$COPYUT1,\$DISKUT2,\$INITDSK,\$COPY

Explanation: You attempted to access either volume \$\$EDXLIB or \$\$EDXVOL, which are reserved for use by the system.

System Action: Returns to command mode.

User Response: Specify a valid volume name and retry the function.

SYMBOL IS MULTIPLY DEFINED

Issued by: \$EDXASM

Explanation: The same 1–8 character symbol has been explicitly defined more than once.

System Action: The error is flagged and compilation continues.

User Response: Ensure that duplicate definitions do not appear in the code.

SYSTEM RTCODE = xxx

Issued by: \$IAMUT1

Explanation: During the secondary index load, an Indexed Access Method error occurred and was reported in a previous message. This message provides further information when the error was caused by the failure of a system function which was issued by the Indexed Access Method.

System Action: System prompts you: RETRY (Y/N)?

User Response: Check the system return code to find the cause of the problem, and take appropriate action.

**TAPE ERROR - CYLINDER xxx, TRACK yyy
NOT RESTORED. RESTORE
PROCEEDING**

Issued by: \$TAPEUT1

Explanation: Using the RT command, the restore utility encountered a read error on the tape. The tape record contained a full track's worth of data. That data has been lost and will not be written to the disk. The error occurred while it was processing cylinder xxx and track yyy on the disk.

System Action: The utility continues.

User Response: The disk is only partially restored. Use \$DISKUT1 (LACTS command) to determine which data sets are affected.

**TAPE ERROR - FIXED HEAD TRACK xxx
NOT RESTORED. RESTORE
PROCEEDING**

Issued by: \$TAPEUT1

Explanation: Using the RT command, the restore utility encountered a read error on the tape while it was restoring the fixed head area on the disk. The tape record contained a full track's worth of data. That data has been lost and will not be written to the disk. The error occurred while it was processing track xxx under the fixed heads.

System Action: The utility continues.

User Response: The fixed head area is only partially restored. Use \$DISKUT1 (LACTS command) to determine which data sets are affected.

**TAPE ERROR RC = xxx
RECORD NUMBER = yyy**

Issued by: \$TAPEUT1

Explanation: During the save of a tape, an error occurred in reading data to the save device. The error occurred at record yyy of the tape. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action. Then retry.

**TAPE xxx IS NOT A TAPE * * * TAPE xxx
MARKED UNUSABLE**

Issued by: \$TAPEUT1

Explanation: An address was incorrectly defined (for example, the device is not a tape); the tape drive was not turned on; or the tape drive has a hardware failure.

System Action: Terminates the initialization.

User Response: Correct the problem and restart the initialization.

TAPE I/O ERROR RC = xxx

Issued by: \$TAPEUT1

Explanation: During the allocate function, an error occurred in either reading or writing to the tape. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action. Then retry the function.

TAPE NOT COMPATIBLE WITH RESTORE UTILITY

Issued by: \$TAPEUT1

Explanation: You cannot use a tape for the restore function. You must use a disk.

System Action: Ends the operation.

User Response: Specify a disk device for use with the restore function.

TAPE xxx OFFLINE FOR BLP yyy BPI

Issued by: \$TAPEUT1

Explanation: The address xxx is valid but the tape is not mounted.

System Action: Terminates the initialization.

User Response: Mount the tape and restart the initialization.

TAPE VOLUMES NOT ALLOWED - ERROR

Issued by: \$COPY

Explanation: You cannot use a tape volume for any \$COPY function.

System Action: Ends the operation.

User Response: Specify a disk or diskette volume for use with \$COPY.

TARGET EDX DATA SET TOO SMALL - MUST REALLOCATE IT

Issued by: \$HXUT1

Explanation: After the RX command was entered, \$HXUT1 determined the target data set was not large enough to contain the multivolume H-exchange data sets.

System Action: The RX request terminates.

User Response: Use \$DISKUT1 to determine the allocation size of the target data set. Delete the target data set and then reallocate the target at a larger size and retry.

TARGET VOLUME IS IN USE BY ANOTHER PROGRAM

Issued by: \$COPY

Explanation: You cannot copy to the specified target volume because it is already in use.

System Action: Ends the operation.

User Response: Specify another target volume or wait and retry with the original target volume.

TASK xxx ENTERED

Issued by: \$BSCUT2

Explanation: In response to your request, the utility is exercising one of the nine read or write tasks on a BSC line.

System Action: Continues processing.

User Response: This is an informational message. No user response is needed.

TASK xxx EXITED

Issued by: \$BSCUT2

Explanation: The utility has completed simulation of one of the nine read or write tasks on a BSC line.

System Action: Continues processing.

User Response: This is an informational message. No user response is needed.

TASK NAME NOT SPECIFIED

Issued by: \$EDXASM

Explanation: A label must be coded on the instruction to define the task name.

System Action: The error is flagged and compilation continues.

User Response: Specify a 1–8 character name as the task name.

TERMINAL xxx BUSY

Issued by: \$MTM

Explanation: Terminal xxx specified in the TERMINAL file is connected to another program.

System Action: Does not use the terminal.

User Response: Try to RECONNECT at a later time.

TERMINAL I/O RC = xxx

Issued by: \$IMAGE

Explanation: A terminal I/O error occurred during the requested function. The result was a return code of xxx.

System Action: Ends the operation.

User Response: Check the terminal I/O general return code to find the cause of the problem, and take appropriate action.

TERMINAL xxx I/O ERROR

Issued by: \$MTM

Explanation: During Multiple Terminal Manager initialization, an I/O error occurred in an attempt to write the IPL screen to the specified terminal (\$TERMERR exit taken). A possible cause is the terminal being powered off at initialization.

System Action: The specified terminal will be in disconnected state.

User Response: Determine cause of the I/O error and take the appropriate action. To reconnect the terminal, use the \$RECON or RECONNECT utilities.

TERMINAL xxx NOT DEFINED IN EVENT DRIVEN EXECUTIVE SYSTEM

Issued by: \$MTM

Explanation: The specified terminal was not included in the definition of terminals when the Event Driven Executive system was generated.

System Action: Does not connect the terminal.

User Response: Include a terminal definition for the specified terminal when the Event Driven Executive system is generated.

TERMINAL NAME INVALID

Issued by: \$MTM

Explanation: The terminal name specified for the TERMINAL file record listed immediately before this message is invalid.

System Action: Does not connect the terminal.

User Response: Correct the TERMINAL record. Stop and restart the Multiple Terminal Manager.

TERMINAL NOT RENAMED TO BLANKS

Issued by: \$TERMUT1

Explanation: You entered blanks as the new terminal name. The utility will not change a terminal name to blanks.

System Action: Returns to command mode.

User Response: Enter a valid name for the new terminal name.

TERMINAL xxx NOT RECONNECTED -- INVALID FOR PSEUDO

Issued by: \$MTM

Explanation: An attempt was made to reconnect a pseudo terminal. There is no physical device to which a pseudo terminal can be connected. A pseudo terminal need not be connected to run Multiple Terminal Manager applications.

System Action: The specified pseudo terminal remains disconnected.

User Response: This is an informational message. No user response is needed.

TERMINAL xxx RECONNECTED

Issued by: \$MTM

Explanation: The named terminal has been reconnected to the Multiple Terminal Manager.

System Action: Reconnects the terminal to the Multiple Terminal Manager.

User Response: Use the terminal as needed.

TERMINAL: xxx IS NOT ON THIS SYSTEM

Issued by: \$TERMUT1, \$JOBQUT

Explanation: The terminal name you specified is not supported.

System Action: Gives logging terminal name and prompts for new name.

User Response: Enter a valid terminal name, an asterisk (*) for the terminal you are using, or just press enter to keep the current name.

TERMINAL TABLE OR STORAGE SIZE EXCEEDED

Issued by: \$MTM

Explanation: While building the terminal table and loading servers, the storage size or the the maximum number of terminals (10) allowed has been exceeded. The work space, defined in CDMINIT, is defined to allow a maximum of 50 terminals.

System Action: Initialization is aborted.

User Response: Increase the terminal table size by changing module CDMCOMMN. If there is not enough room, make the partition larger, decrease the number of terminals, or make CDMDUMMY smaller. Restart the Multiple Terminal Manager.

**TERMINAL VARYOFF UNSUCCESSFUL -
TERMINAL CODE = xxx**

Issued by: \$TERMUT1

Explanation: An error occurred which prevented the varyoff.

System Action: Returns to command mode.

User Response: Check the terminal I/O general return code to find the cause of the problem, and take appropriate action.

**TERMINAL VARYON UNSUCCESSFUL -
RC = xxx**

Issued by: \$TERMUT1

Explanation: An error occurred which prevented the varyon.

System Action: Returns to command mode.

User Response: Check the terminal I/O general return code to find the cause of the problem, and take appropriate action.

**TEXT MESSAGE IS NOT A VALID
CHARACTER STRING**

Issued by: \$EDXASM

Explanation: Valid text messages in the form of character strings must be enclosed in quotes.

System Action: The error is flagged and compilation continues.

User Response: Enclose the text message in quotes.

THAT IS NOT A VALID JOB NUMBER

Issued by: \$SUBMIT

Explanation: You specified a job number that does not exist.

System Action: Returns to command mode.

User Response: Obtain the correct job number (STQ command) and retry.

THE ?? DATA SET NAMES CANNOT BE PRELOCATED

Issued by: \$PREFIND

Explanation: You must supply the actual data set names and not just “??”.

System Action: Ends the operation.

User Response: Enter the actual data set names and retry the function.

THE ?? NAMES CANNOT BE RESOLVED IN JOB MODE

Issued by: \$PREFIND

Explanation: You did not specify the actual data set names to be opened.

System Action: Ends the operation.

User Response: Specify the actual data set names and retry the function.

THE FIRST PI ENTRY IS INVALID

Issued by: \$EDXASM

Explanation: The first address specified on the PI operand of the SENSORIO statement must be equal to the base address or base address+1 (as determined by the ADDRESS= operand).

System Action: The error is flagged and compilation continues.

User Response: Code the first PI entry as explained.

THERE IS NO FORMAT TO PRINT

Issued by: \$IMAGE

Explanation: There is no screen image defined yet.

System Action: Ends the operation.

User Response: Define the screen image format in edit mode.

THERE ARE xxx RECORDS IN YOUR DEVICE. THE DEFAULT OF THE VOLUME DIRECTORY FOR THIS DEVICE IS RECORD yyy. IT NOW EXISTS AT RECORD zzz.

Issued by: \$INITDSK

Explanation: The utility is asking if you want to change the location on disk of the device directory.

System Action: Awaits operator response.

User Response: Press the enter key for the default location or enter the new location.

THE ROW OR COLUMN FROM AN SBA WAS IN ERROR

Issued by: \$IMAGE

Explanation: The utility could not find a 3101 character in the 3101 translation table.

System Action: Ends the operation.

User Response: Correct the image to eliminate invalid characters and retry.

THE SAVE WAS CANCELLED BEFORE IT COMPLETED

Issued by: \$TAPEUT1

Explanation: Using the RT command, the save utility (\$TAPEUT1) was cancelled before it completed saving the entire device, volume or data set.

System Action: Terminates the utility.

User Response: The target device, volume, or data set is only partially restored. Inspect the data (via \$DISKUT2 DP command, for example) to see if it is sufficient. If not, delete the volume or data set. A partial device restore may be acceptable if the part not restored is never accessed.

THE SECOND PI ENTRY IS INVALID

Issued by: \$EDXASM

Explanation: The second address specified on the PI operand of the SENSORIO statement must be equal to the base address or base address+1 (as determined by the ADDRESS= operand).

System Action: The error is flagged and compilation continues.

User Response: Code the second PI entry as explained.

THE TOTAL NUMBER OF OPERAND DELIMITERS EXCEEDS MAXIMUM (70)

Issued by: \$EDXASM

Explanation: There can only be a maximum of 70 delimiters in an instruction. A delimiter is either a left parenthesis, right parenthesis, or a comma.

System Action: The error is flagged and compilation continues.

User Response: Ensure the delimiters do not exceed the maximum of 70.

THE TWO PI ENTRIES ARE EQUAL

Issued by: \$EDXASM

Explanation: The two process interrupt entries specified on the PI= operand are the same. (SENSORIO statement)

System Action: The error is flagged and compilation continues.

User Response: Examine the PI= entries and eliminate all duplicate addresses.

T

THIS ADDRESS HAS BEEN PREVIOUSLY DEFINED

Issued by: \$EDXASM

Explanation: An address specified by one of the operands on this SENSORIO statement has been previously defined on this or a previously defined SENSORIO statement.

System Action: The error is flagged and compilation continues.

User Response: Check the addresses specified and ensure there are no duplicate addresses.

THISBUF NAME TOO LONG - CAREAD TERMINATED

Issued by: \$\$1ASM

Explanation: The thisbuf operand on the CAREAD instruction has more than eight characters.

System Action: Terminates the CAREAD instruction.

User Response: On the CAREAD instruction, specify a thisbuf operand that is less than or equal to eight characters in length, and reassemble.

THISBUF ADDR IS NUMERIC - CAREAD TERMINATED

Issued by: \$\$1ASM

Explanation: The thisbuf operand on the CAREAD instruction is numeric. The first character must be alphabetic.

System Action: Terminates the CAREAD instruction.

User Response: On the CAREAD instruction, specify a valid thisbuf operand whose first character is alphabetic, and reassemble.

THISBUF NAME IS NUMERIC - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The thisbuf operand on the CAREAD instruction is numeric. The first character must be alphabetic.

System Action: Terminates the expansion.

User Response: On the CAREAD instruction, specify a valid thisbuf operand whose first character is alphabetic, and reassemble.

THISBUF NOT SPECIFIED - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The thisbuf operand on the CAREAD instruction is a required operand and is not specified.

System Action: Terminates the expansion.

User Response: On the CAREAD instruction, specify a valid thisbuf operand, and reassemble.

THIS CPU UNABLE TO IPL

Issued by: \$\$S1S1UT1

Explanation: Your processor is assigned as the secondary processor (RPQ D02242 is installed) and cannot perform IPL of the primary.

System Action: None.

User Response: This is an informational message. No user response is needed.

THIS MEMBER NOT A DATA MEMBER

Issued by: \$DIUTIL

Explanation: The data set name you specified refers to a program-type data set. You must specify a data-type data set.

System Action: Ends the operation.

User Response: Reenter the information and supply the name of a data-type data set.

THIS NUCLEUS WAS CONFIGURED WITHOUT THE DEBUG OPTION

Issued by: \$DEBUG

Explanation: During system generation, the required support for the \$DEBUG utility was not included. You cannot perform the debug function on your system.

System Action: Ends the operation.

User Response: To include the \$DEBUG utility, you must perform another system generation and include the required supervisor modules. Refer to the *Installation and System Generation Guide* for details on this procedure.

TIMERS= REQUIRES 4 DECIMAL VALUES

Issued by: \$EDXASM

Explanation: If the TIMERS= operand is specified on the ACCA TERMINAL statement, all four timer values must be specified. The timer defaults for the LMODE= type are assumed if TIMERS= is omitted.

System Action: The error is flagged and compilation continues.

User Response: Specify the TIMERS= operand with four values or assume the LMODE= default. Refer to the *Installation and System Generation Guide* for details on how to specify a TERMINAL statement for ACCA devices.

T

TITLE NAME IS NUMERIC - CAPRINT TERMINATED

Issued by: \$S1ASM

Explanation: The TITLE= operand on the CAPRINT instruction is numeric. The first character must be alphabetic.

System Action: Terminates the CAPRINT instruction.

User Response: On the CAPRINT instruction, specify a valid TITLE= operand whose first character is alphabetic, and reassemble.

TITLE NAME IS NUMERIC - EXPANSION TERMINATED

Issued by: \$EDXASM

Explanation: The TITLE= operand on the CAPRINT instruction is numeric. The first character must be alphabetic.

System Action: Terminates the expansion.

User Response: On the CAPRINT instruction, specify a valid TITLE= operand whose first character is alphabetic, and reassemble.

TITLE NAME TOO LONG - CAPRINT TERMINATED

Issued by: \$S1ASM

Explanation: The TITLE= operand on the CAPRINT instruction has more than eight characters.

System Action: Terminates the CAPRINT instruction.

User Response: On the CAPRINT instruction, specify a TITLE= operand that is less than or equal to eight characters in length, and reassemble.

TO KEY SPECIFIED WITH #1 OR #2

Issued by: \$EDXASM

Explanation: Index register #1 or #2 cannot be specified as a destination address (operand 1) in a cross partition move.

System Action: The error is flagged and compilation continues.

User Response: Recode the cross partition move without using an index register as the destination address.

TOO FEW PREC= SPECIFICATIONS

Issued by: \$EDXASM

Explanation: Fewer than the minimum required parameters were specified as the precision for floating-point arithmetic on the PREC= operand.

System Action: The error is flagged and compilation continues.

User Response: If RESULT= is coded, three precision parameters are required; otherwise specify two precision parameters.

TOO MANY COPIES OF \$DEBUG ARE CURRENTLY ACTIVE

Issued by: \$DEBUG

Explanation: Only a limited number of copies of the debug utility can be executing at the same time. You cannot use \$DEBUG at this time.

System Action: Ends the operation.

User Response: To find out where all the copies of \$DEBUG are executing, enter the \$A ALL command. You may have to wait until someone is done using \$DEBUG before you can try to load it again.

TOO MANY DATA SETS SPECIFIED

Issued by: \$EDXASM

Explanation: The number of data sets specified for the DS= operand of the PROGRAM statement exceeds the limit of nine.

System Action: The error is flagged and compilation continues.

User Response: Specify no more than nine data sets for the DS= operand.

TOO MANY OVERLAY PROGRAMS SPECIFIED

Issued by: \$EDXASM

Explanation: The number of overlay programs specified by the PGMS= operand of the PROGRAM statement exceeds the maximum of nine.

System Action: The error is flagged and compilation continues.

User Response: Specify no more than nine programs for the PGMS= operand.

TOO MANY PI= ENTRIES

Issued by: \$EDXASM

Explanation: A maximum of two addresses can be specified in the PI operand of the SENSORIO statement.

System Action: The error is flagged and compilation continues.

User Response: Ensure that the PI operand has no more than two entries.

**TOO MANY POSITIONAL OPERANDS
WERE SPECIFIED****Issued by:** \$EDXASM**Explanation:** More positional operands were specified than are valid for the instruction.**System Action:** The error is flagged and compilation continues.**User Response:** Check the syntax and description of the instruction and code it accordingly.**TOTAL SIZE LESS THAN THE MINIMUM
OF 4K****Issued by:** \$IAMUT1**Explanation:** Using the PP command, the sum of the sizes of the page areas in all partitions was less than the minimum required for data paging.**System Action:** Page area sizes were not modified.**User Response:** Either assign more storage for data paging or do not attempt to use it.**TRACE SIZE TOO LARGE - EXPANSION
TERMINATED****Issued by:** \$EDXASM**Explanation:** The TRSIZE= operand on the CACBx statement contains a value greater than 'FF'.**System Action:** Terminates the expansion.**User Response:** On the CACBx statement, specify a TRSIZE= operand that is less than or equal to 'FF', and reassemble.**TRACE SIZE TOO SMALL - EXPANSION
TERMINATED****Issued by:** \$EDXASM**Explanation:** The TRSIZE= operand on the CACBx statement contains a value less than zero.**System Action:** Terminates the expansion.**User Response:** On the CACBx statement, specify a TRSIZE= operand that is greater than or equal to zero, and reassemble.**\$TRAP MUST BE IN PARTITION #1. \$TRAP
TERMINATED****Issued by:** \$TRAP**Explanation:** \$TRAP was loaded into a partition other than partition 1. It must be loaded into partition 1.**System Action:** Terminates \$TRAP.**User Response:** Load \$TRAP into partition 1 and restart the program.

TRSIZE SUPPLIED IS NOT NUMERIC

Issued by: \$S1ASM

Explanation: The TRSIZE= operand on the CACB1 statement is not numeric.

System Action: Terminates the CAIOCB statement.

User Response: On the CACB1 statement specify a valid TRSIZE= operand that is numeric, and reassemble.

**TRSIZE SUPPLIED IS NOT NUMERIC -
EXPANSION TERMINATED**

Issued by: \$EDXASM

Explanation: The TRSIZE= operand on the CACBx statement is not numeric.

System Action: Terminates the expansion.

User Response: On the CACBx statement, specify a valid TRSIZE= operand that is numeric, and reassemble.

**TWOKBLK KEYWORD MAY NOT
EXCEED 64**

Issued by: \$EDXASM

Explanation: For the STORBLK instruction, the maximum value coded for the TWOKBLK operand may not exceed 64.

System Action: Continues processing but does not generate object text for that instruction.

User Response: Correct the indicated error and reassemble.

**TYPE COUNT MUST BE BETWEEN 0 AND
255**

Issued by: \$EDXASM

Explanation: The count specified for the TERMCTRL GETSTORE/PUTSTORE function is not between 0 and 255 bytes.

System Action: The error is flagged and compilation continues.

User Response: Specify the value for count between 0 and 255.

TYPE=DSECT IS NOT SUPPORTED

Issued by: \$EDXASM

Explanation: TYPE=DSECT cannot be specified on the TERMINAL statement of a program compiled by \$EDXASM.

System Action: The error is flagged and compilation continues.

User Response: Delete this operand from your program's TERMINAL statement.

TYPE=MT REQUIRES POLL= WITH 1 - 4 ADDRESSES

Issued by: \$EDXASM

Explanation: TYPE=MT was specified on the BSCLINE statement but the required POLL= operand was omitted.

System Action: The error is flagged and compilation continues.

User Response: Specify a list of 1 - 4 poll/select addresses as the POLL operand. Each address consists of two hexadecimal digits.

TYPE= MUST BE 'STG' OR 'DSK'

Issued by: \$EDXASM

Explanation: You coded an invalid parameter on the TYPE= operand of the COMP statement. Specify either STG or DSK.

System Action: The error is flagged and compilation continues.

User Response: Code a valid parameter for the TYPE= operand and recompile the program.

UNABLE TO ALLOCATE DATA SET xxx,yyy

Issued by: \$MSGUT1

Explanation: The utility could not allocate the data set.

System Action: Returns to command mode.

User Response: Compress the volume in use or specify the use of another volume. Then retry the function.

UNABLE TO ASSIGN ALTERNATE CYLINDER

Issued by: \$DASDI

Explanation: An attempt failed to assign an alternate sector for a defective cylinder.

System Action: Returns to command mode.

User Response: Device is unusable. Use another device.

UNABLE TO DELETE DATA SET xxx

Issued by: \$MSGUT1

Explanation: You cannot use \$MSGUT1 to delete data sets in this particular case.

System Action: Returns to command mode.

User Response: Use \$DISKUT1 to delete the data set, then retry the function.

UNABLE TO LOAD REQUESTED PROGRAM

Issued by: \$DEBUG

Explanation: The name you specified refers to a data set and not to a program, or to a program that does not exist.

System Action: Ends the operation.

User Response: Specify the name of a valid, loadable program.

UNABLE TO LOAD REQUESTED TAPE FUNCTION RC = xxx

Issued by: \$TAPEUT1

Explanation: The requested program cannot be loaded for the tape function in progress.

System Action: If the tape function is ST, RT, CD, or EX, the utility terminates. Otherwise, it returns to command mode.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

U

UNABLE TO READ SECONDARY INDEX FCB, READ RTCODE= xxx

Issued by: \$IAMUT1

Explanation: During the secondary index load, an error occurred while attempting to read the first record of the secondary index file. The result was a return code of xxx.

System Action: System prompts you: RETRY (Y/N)?

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

UNBALANCED PARENTHESIS IN OPERAND

Issued by: \$EDXASM

Explanation: All open parentheses must have corresponding close parentheses and vice versa.

System Action: The error is flagged and compilation continues.

User Response: Ensure the parentheses are all paired correctly.

UNBALANCED QUOTES IN OPERAND

Issued by: \$EDXASM

Explanation: No terminating quote is present in the character string or quotes within the character string are not specified correctly.

System Action: The error is flagged and compilation continues.

User Response: Ensure that all beginning and ending quotes are paired correctly.

UNDEFINED ERROR FROM \$IMOPEN RC = xxx

Issued by: \$IMAGE

Explanation: An I/O error occurred resulting in a return code from \$IMOPEN.

System Action: Ends the operation.

User Response: Check the \$IMOPEN return code to find the cause of the problem, and take appropriate action.

UNEXPIRED DATA SET! CONTINUE?

Issued by: \$TAPEUT1

Explanation: The tape label data has not expired.

System Action: If answer is "no", returns to command mode. If the answer is "yes", continues processing.

User Response: Answer the prompt "Y" to overwrite the expiration.

**WARNING - VOLUME SEQUENCE
NUMBER MUST BE BLANK OR 01 THRU
99, IT IS xxx**

Issued by: \$HXUT1

Explanation: The volume sequence number indicator does not contain a blank (no volume sequence checking) or 01 through 99.

System Action: The utility issues the message "CONFLICT WITH H EXCHANGE STANDARD".

User Response: Enter the UD command (update data set label by name) or the UH command (update the data set label by number) and specify a blank or the volume sequence number (01-99) to the prompt for the multivolume indicator.

**WARNING - VOLUME SURFACE
INDICATOR MUST BE M, IT IS xxx**

Issued by: \$HXUT1

Explanation: The volume surface indicator of the H-exchange diskette does not contain an "M" (indicates two double-density surfaces).

System Action: The utility issues the message "CONFLICT WITH H EXCHANGE STANDARD" and does one of the following:

- Rejects the diskette, if you have just loaded \$HXUT1 and specified the volume name, or you issued a CV command (changed to another volume).
- Continues to prompt for the required volume label field until it is entered. This occurs only when you issue a UV command and update the volume label inconsistent with H-exchange standards.

User Response: If the diskette is rejected, re-create the diskette consistent with H-exchange standards, otherwise (with the UV command) specify an "M" to the prompt for the volume surface indicator.

W

WARNING - UNRESOLVED WEAK EXTERNAL REFERENCES

Issued by: \$EDXLINK

Explanation: One or more weak external references were not resolved.

System Action: Generates a list of all unresolved weak external references and then continues. This will not affect the completion code.

User Response: If you see any weak external references in the list you wanted to have resolved, include the module necessary to resolve them. Then relink.

WARNING - WRITE PROTECT INDICATOR IS NOT BLANK OR P, IT IS xxx

Issued by: \$HXUT1

Explanation: The write protect indicator of the H-exchange diskette does not contain a blank (allows read/write access) or "P" (read only access).

System Action: The utility issues the message "CONFLICT WITH H EXCHANGE STANDARD".

User Response: Enter the UD command (update data set label by name) or the UH command (update the data set label by number) and specify a blank or "P" to the prompt for write protect.

WORKBUFFER OVERFLOW ENCOUNTERED ON LINE #xxx

Issued by: \$MSGUT1

Explanation: You did not enter an end character or continuation character for the message on that line.

System Action: Continues processing.

User Response: When processing finishes, correct the error and retry.

WORK DATA SET NOT FOUND ON DISK - ERROR EXIT

Issued by: \$COPY

Explanation: The data set you specified does not exist on the disk.

System Action: Returns to command mode.

User Response: Specify the name of a valid disk data set and retry the function.

WRITE ERROR OCCURRED, RC = xxx

Issued by: \$XPSLINK from \$XPSPOST

Explanation: While attempting to write a record to disk or diskette, an error occurred. The result was a return code of xxx.

System Action: \$XPSPOST ends unsuccessfully with a completion code of 12. The nucleus is not stored.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**WRITE ERROR ENCOUNTERED ON xxx,yyy
RC = zzz**

Issued by: \$XPSLINK from \$XPSPRE

Explanation: \$XPSPRE was unable to write to the specified data set. The result was a return code of zzz.

System Action: Ends the operation.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**WRITE ERROR ON RECORD NUMBER xxx
RC = yyy**

Issued by: \$COPY

Explanation: An error occurred while writing the specified record to disk.

System Action: Returns to command mode.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

**WRITE ERROR ON UPDATED PROGRAM
HEADER**

Issued by: \$PREFIND

Explanation: A disk write error occurred while prefinding a program header.

System Action: Ends the operation.

User Response: Link and/or update the program, and retry the prefind function.

**WRITE ERROR - RC = xxx ENCOUNTERED
AT APPROXIMATELY yyy FEET OF
TAPE**

Issued by: \$TAPEUT1

Explanation: An error occurred in writing records at the tape length shown.

System Action: Continues processing.

User Response: Check the tape read/write return code to find the cause of the problem, and take appropriate action.

**WRITE OUTPUT DATASET RETURN CODE
= xxx**

Issued by: \$IAMUT1

Explanation: Using the RO or UN commands, an error occurred when writing to the output data set. The result was a return code of xxx.

System Action: If the command is RO, the function is terminated. If the command is UN, you are prompted for another output data set.

User Response: Check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action.

WRITE OUTPUT DATASET RETURN CODE

= xxx. RECORD NUMBER = yyy.

Issued by: \$IAMUT1

Explanation: An attempt to write to a sequential (output) data set failed. The result was a return code of xxx that occurred at record yyy.

System Action: Terminates command.

User Response: For the LO or RO commands, check the Indexed Access Method return code to find the cause of the problem, and take appropriate action. (Note 1.) For the UN command, check the disk and diskette read/write return code to find the cause of the problem, and take appropriate action. (Note 2).

Notes:

1. It may be necessary to redefine data set (SE,DF) and retry LO command.
2. It may be necessary to reallocate the data set and retry the UN command.

WRITER STOPPED

Issued by: Spooling

Explanation: The specified spool writer has been stopped.

System Action: Any spool jobs targeted to the printer to which the stopped writer was assigned remain unprinted.

User Response: Restart the spool writer as needed.

WRITER LOAD ERROR, RC=xxx

Issued by: Spooling

Explanation: An attempt was made to load a writer into storage and the load failed. The result was a LOAD return code of xxx.

System Action: Does not start the spool writer.

User Response: Check the LOAD return code to find the cause of the problem and take the appropriate action.

XD= MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: The parameter specified for the XD= operand of the DCB statement must be either YES or NO.

System Action: The error is flagged and compilation continues.

User Response: Specifying XD= YES indicates the DCB is a non-standard type. XD=NO indicates a standard DCB and is the default.

XLATE= MUST BE YES OR NO

Issued by: \$EDXASM

Explanation: On a message instruction, the operand XLATE= must be either YES or NO.

System Action: The error is flagged and compilation continues.

User Response: After the compile completes, correct the error and rerun.

**XPSSTK MUST BE BETWEEN 10 AND 128 --
DEFAULT IS 20**

Issued by: \$EDXASM

Explanation: During system generation, you coded a value outside the valid range of 10–128 for the XPSSTK operand of the SYSTEM statement.

System Action: The error is flagged and compilation continues.

User Response: Code a value from 10–128 for the XPSSTK operand and recompile the system definition statements.

**X.21 SW CONNECTION FAILED FOR
DEVICE ADDRESS xxx
RETURN CODE = yyy**

Issued by: \$SNAMAIN

Explanation: The X.21 Circuit Switched Network is unable to complete your call request. Device address xxx is in decimal.

System Action: SNA network deactivates and issues the return code yyy.

User Response: Check the X.21 Circuit Switched Network return codes for a description of the code returned in this message. Correct the problem and reactivate the SNA network.

X.21 SW REDIRECTION ACTIVATED FOR DEVICE ADDRESS xxx

Issued by: \$SNAMAIN

Explanation: The X.21 Circuit Switched Network has processed your redirection request. Device address xxx is in decimal.

System Action: SNA network deactivates.

User Response: To continue processing, you can:

- Redefine the SNA network, or
- Change the X.21 connection record for the SNA network and reactivate the network.

X.21 SW REGISTRATION/CANCELLATION CONFIRMED FOR DEVICE ADDRESS xxx

Issued by: \$SNAMAIN

Explanation: The X.21 Circuit Switched Network has processed your registration or cancellation request. Device address xxx is in decimal.

System Action: SNA network deactivates.

User Response: To continue processing, you can:

- Redefine the SNA network, or
- Change the X.21 connection record for the SNA network and reactivate the network.

X.21 SW REDIRECTION DEACTIVATED FOR DEVICE ADDRESS xxx

Issued by: \$SNAMAIN

Explanation: The X.21 Circuit Switched Network has processed your redirection deactivation request. Device address xxx is in decimal.

System Action: SNA network deactivates.

User Response: To continue processing, you can:

- Redefine the SNA network, or
- Change the X.21 connection record for the SNA network and reactivate the network.

X.21 SW TEST FAILED FOR DEVICE ADDRESS xxx RETURN CODE = yyy

Issued by: \$SNAMAIN

Explanation: The 2080 feature card is jumpered incorrectly or is malfunctioning. Device address xxx is in decimal.

System Action: SNA network deactivates and issues the return code yyy.

User Response: Check the X.21 Circuit Switched Network return codes for a description of the code returned in this message. Correct the problem and reactivate the SNA network. (Refer to the *IBM Implementation of X.21 Interface General Information Manual*, GA27-3287 for information on how to jumper the 2080 feature card for the X.21 Circuit Switched Network.)

#1 OR #2 USED IN FROM KEY OR TO KEY

Issued by: \$EDXASM

Explanation: Index registers #1 and #2 cannot be used as addresses in a cross partition move.

System Action: The error is flagged and compilation continues.

User Response: Recode the operation so that an index register is not used as a source or destination address.

4975-02L REQUIRES BITRATE OF 4800

Issued by: \$EDXASM

Explanation: The bit rate for the 4975-02L printer must be 4800 bps.

System Action: The error is flagged and compilation continues.

User Response: Specify BITRATE=4800. If omitted, 4800 bps is the default.

4975-01L REQUIRES BITRATE OF 2400

Issued by: \$EDXASM

Explanation: The bit rate for the 4975-01L printer must be 2400 bps.

System Action: The error is flagged and compilation continues.

User Response: Specify BITRATE=2400. If omitted, 2400 bps is the default.

Part 2. Codes

This part presents several types of codes issued by the Event Driven Executive:

- Completion** Issued by a utility program or \$EDXASM upon completion to indicate whether or not execution was successful.
- Post** Issued by the system to signal the occurrence of an event. Unless otherwise stated, post codes are returned in the first word of the event control block (ECB) that is posted when the event occurs. You must specify the ECB to be posted with the ECB statement.
- Return** Issued as the result of executing an Event Driven Language instruction or subroutine to indicate whether an operation was a success or failure. The instruction can be in an application program or in a utility. Most return codes are returned in the first word of the task control block (TCB) of the program or task issuing the instruction. The TCB begins at the taskname (label) you specify on the PROGRAM or TASK statement.
- Stop** Issued on Series/1 programmer console to indicate an error condition while the system is running in diagnostic mode, and therefore cannot issue error messages or return codes.

The completion and post codes are presented alphabetically, according to the function that issues them.

The return codes are arranged in two ways: by *the function that issues them*, and by *number*. When you get a return code, you can look it up by its number and also by the issuing function. Many codes are issued by several functions.

Part 2 consists of the following chapters:

- Chapter 3, “Completion Codes” lists and explains completion codes, which indicate the completion status of a utility or \$EDXASM.
- Chapter 4, “Post Codes” lists and explains post codes, which signal the occurrence of events while a program is executing.
- Chapter 5, “Return Codes” lists and explains return codes, which indicate the execution status of an Event Driven Language instruction or subroutine.
- Chapter 6, “Stop Codes” lists and explains stop codes, which signal error conditions when the system is running in diagnostic mode and cannot issue messages or return codes.

Chapter 3. Completion Codes

The completion codes and their meanings are presented in alphabetic order as follows:

- `$EDXASM`
- `$EDXLINK`
- `$JOBUTIL`
- `$UPDATE`
- `$XPSLINK`, `$XPSPOST`, `$XPSPRE`

The completion codes are printed by the program on the specified list device upon completion unless otherwise noted.

A completion code of zero is returned if the program is cancelled.

Completion Codes

\$EDXASM Completion Codes

\$EDXASM completion codes are accompanied by an appropriate error message and appear at the end of the \$EDXASM listing. The completion codes can be tested by the job stream processor, allowing steps subsequent to the assembly to be skipped, if appropriate. The completion codes are:

Completion Code	Condition
-1	Successful completion - no errors in assembly
4	Successful completion - one or more statements had warning type assembly errors that did not effect the generation of the object module data set
8	Unsuccessful completion - one or more statements had assembly errors that must be corrected
12	Location counter error - program size exceeds 64K
12	Out of space in work or object data set
12	I/O error in source, work, or object data set
12	Overlay-instruction table full
12	Unable to locate overlay program or copy code module
100	Operator cancelled assembly with ATTN CA command

\$EDXLINK Completion Codes

The completion codes appear at the end of the \$EDXLINK listing.

<u>Completion Code</u>	<u>Condition</u>
-1	Successful completion
4	Warning - possible control statement syntax error or one or more unresolved EXTRNs. Program was generated and may work.
8	Error occurred. Although a program was generated, you should resolve the problem and re-link.
12	Severe error occurred and no program was generated. Resolve problem and re-link.
16	Terminal error occurred. \$EDXLINK is unable to complete normal processing

Completion Codes

\$JOBUTIL Completion Codes

The \$JOBUTIL completion codes are displayed on the terminal from which \$JOBUTIL was loaded.

Completion Code	Condition
-1	Successful completion
61	The transient loader (\$LOADER) is not included in the system
64	No space available for the transient loader
67	A disk or diskette I/O error occurred during the load process
70	Not enough main storage available for the program
71	Program not found on the specified volume
72	Disk or diskette I/O error while reading directory
73	Disk or diskette I/O error while reading program header
74	Referenced module is not a program
75	Referenced module is not a data set
76	Data set not found on referenced volume
77	Invalid data set name
78	LOAD instruction did not specify required data set(s)
79	LOAD instruction did not specify required parameter(s)
80	Invalid volume label specified; for example, greater than eight characters

\$UPDATE Completion Codes

The text of the \$UPDATE completion codes is written to the device specified to receive the printed output. The completion codes can be tested in a \$JOBUTIL job stream.

Completion Code	Condition
-1	Successful completion
8	No supervisor space in this library
8	Output name specified is not a program
8	Disk volume already in use by another program
8	No space in directory
8	No space in data set (output library)
8	Invalid header format
8	Invalid program name
8	Volume not mounted
8	Volume off line
8	Library not found
8	Input data set not found
8	No parameter supplied via \$JOBUTIL
8	No data set names provided via \$JOBUTIL
8	Replacement of output data set not allowed
12	Any disk or diskette I/O errors

Completion Codes

\$XPSLINK, \$XPSPOST, \$XPSPRE Completion Codes

The cross-partition services utilities (\$XPSLINK, \$XPSPOST, and \$XPSPRE) issue completion codes to the terminal where the utility was invoked.

Completion Code	Condition
-1	Successful completion
i	Minor error; statement ignored, processing continues
8	Major error; \$XPSPRE ends; \$XPSLINK loads \$XPSPOST to delete any temporary data sets allocated by \$XPSPRE
12	Severe error; utility ends

Chapter 4. Post Codes

The post codes and their meanings are presented as follows:

- BIND event (SNA)
- Channel Attach
- Tape
- WAITM instruction

The Event Driven Language and function post codes are issued to signal the occurrence of an event. Unless otherwise stated, the post codes are returned to the first word of the event control block (ECB) of the calling program.

Post Codes

BIND Event Post Codes

The SNA BIND event post codes are returned to the area specified by the ERRCODE parameter of the NETINIT instruction.

Post Code	Condition
2	UNBIND HOLD received
1	UNBIND received
-9	Session terminated
-10	Non-sequenced procedure error received
-11	Minus response to INITSELF or TERMSELF received

Channel Attach Post Codes

The channel attach post codes are returned to the event control block (ECB) of the calling program for start and stop or to the CAIOCB for open, close, read, and write.

Post Code	Condition
501	EXIO Error - device not attached. The EXIO did not find a channel attach device at the specified address. If the address is correct, this indicates a hardware error.
502	EXIO Error - busy. This is returned by EXIO when the channel attach device reports a busy condition. This occurs most frequently during multiple port applications. If the device is legitimately busy with another port, you can reissue the I/O. Otherwise this indicates a hardware error. Verify that the channel attach device has a shared unit control word (UCW) on the System/370 channel. An incorrect UCW specification can cause this error.
503	EXIO Error - busy after reset. This indicates that a Series/1 reset was detected while the Series/1 and System/370 were connected (had matching commands in the channel attach device). This state is exited when the status is accepted by the System/370, a System/370 reset occurs, or a power-on reset in the Series/1 occurs.
504	EXIO Error - command reject. This occurs when the channel attach device is unable to accept a command from the Series/1, or when there is a hardware error or the attachment has received a different command from the System/370. If there is no hardware problem, try reissuing the I/O after the conflicting System/370 command has been processed by the channel attach program.
505	EXIO Error - intervention required. This means the channel attach device is not powered on and/or enabled.
506	EXIO Error - interface data check. This error should not be returned unless there is a hardware error.
507	EXIO Error - controller busy. This is not a valid condition code for the channel attach device. If returned, a hardware error is assumed.
508-516	These EXIO errors should not occur unless there is a problem with the channel attach software support
520	Interrupt error condition. An attention or exception interrupt was received when a device end or device end/attention interrupt was expected. Examine the trace area to determine the cause of the incorrect interrupt.

Post Codes

Channel Attach Post Codes (*continued*)

Post Code	Condition
521	Negative acknowledgement. This occurs when the System/370 issues a write command as an acknowledgement of the previous Series/1 read, and bit 6 of the first byte of data is off. Response is determined by the retry logic in your application program.
522	Buffer overlay. The System/370 issued two or more write commands to a buffer before the Series/1 issued a read command to read the data. Your application program should be written so that read and write commands are acknowledged and remain in synchronization to prevent this from occurring.
523	Protocol error. A Series/1 program and a System/370 program are both attempting to read or write, or the System/370 issued an EAU when the Series/1 user has issued a write. Check your application programs to determine the cause of the protocol error.
524	Timeout. The Series/1 channel attach support did not respond fast enough (within 480 milliseconds) to a System/370 I/O request.
525	Wrong device type. The device at the user-specified address is not a channel attach device.
567	System error - CAPGM terminating. The CAPGM task error exit module is called due to detection of an error in the channel attach code. Channel attach support is unloaded as part of the error cleanup.
572	Terminal busy error. Returned when a message (such as a system error) needs to be sent to a terminal, but that terminal is owned by another task.
599	CAPGM ended. Returned to a CASTOP instruction or STOP utility command if the channel attach support program is being unloaded. This occurs if no other channel attach devices are active.

Tape Post Codes

The tape post codes are returned to the event control block (ECB) of the calling program.

Post Code	Condition
-1	Function successful
101	TAPEID not found
102	Device not offline
103	Unexpired data set on tape
104	Cannot initialize BLP tapes

Post Codes

WAITM Instruction Post Codes

The WAITM instruction post codes are returned in the first word of the MECB for the instruction.

Post Code	Condition
X'FFFF'	Successful completion
X'BAD0'	WAITM instruction not supported (SWAITM module not in system)
X'BAD1'	Too many WAITM operations active in system (maximum is 20)
X'BAD2'	Cannot reset MECB because another program is using it
X'BAD3'	Invalid number of events specified

Chapter 5. Return Codes

The return codes and their meanings are presented in two ways: by the function that issues them, and by number. The functional list is in alphabetical order, as follows:

- Binary Synchronous Communications
- Channel Attach
- Data Formatting
- Disk and Diskette Read/Write
- \$DISKUT3
- EXIO
- Floating-point
- Formatted Screen Image:
 - \$IMDATA subroutine
 - \$IMOPEN subroutine
 - \$IMPROT subroutine
- General Purpose Interface Bus (GPIB)
- Indexed Access Method
- Job Queue Processor
- LOAD
- Message Handler
- Multiple Terminal Manager
- \$PDS
- SBIO (Sensor-Based I/O)
- Series/1-to-Series/1 Attachment
- SDLC Communications
- Spool
- Systems Network Architecture (SNA)
- Systems Network Architecture extended error codes

Return Codes

- Terminal I/O:
 - General
 - ACCA
 - Interprocessor Communications
 - Virtual Terminal
 - 4975 Printer
- TP (Host Communication Facility)
- Unmapped Storage
- X.21 Circuit Switched Network

The return codes are issued by EDL instructions in application programs, and by utility programs that invoke EDL instructions. In most cases, they are returned in the first word of the task control block of the calling application program, or in messages issued by utilities to terminals and other output devices.

Return Codes (by Function)

Binary Synchronous Communications (BSC) Return Codes

The BSC return codes are placed in the first word of the task control block of the program issuing the BSC instruction. The following utilities also issue BSC return codes to the terminal or the printer in use: \$BSCTRCE, \$BSCUT1, \$BSCUT2, \$RMU, \$RJE2780, and \$RJE3780.

Return Code	Condition	Notes
-32	System is unable to find X.21 support. Re-IPL the system.	7
-31	Not enough storage available to handle the number of X.21 requests. Use the \$DISKUT2 SS command to allocate more storage for \$X21. You can issue three simultaneous requests for every 256 bytes of storage allocated.	7
-30	Your supervisor does not contain X.21 support.	7
-29	System does not have enough storage available to load the X.21 support or the connection record data set, \$\$X21DS, is not on the IPL volume.	7
-27	Unrecoverable hardware error. If \$LOG is active, check the error log record for the X.21 device for more details.	7
-25	Connection failed	7
-24	Time expired for the completion of a call request. Call request failed.	7
-23	You cancelled a call request with a \$C command.	7
-22	Call request failed due to Public Data Network problems. Call progress signals invalid.	7
-21	Call request failed due to Public Data Network problems. Call progress signals incomplete	7
-20	Call request failed and network would not allow request to be retried. If \$LOG is active, check the error log record for the X.21 device for more details.	7
-19	Number of retries exhausted for the call request. If \$LOG is active, check the error log record for the X.21 device for more details.	7
-18	Hardware error for the 2080 feature card. I/O request could not be completed.	7
-16	The Network information field of the X.21 connection record has no plus sign or just a plus sign.	7
-15	The value in the Retry or Delay field of the X.21 connection record exceeds the maximum value allowed.	7
-14	The Retry or Delay field of the X.21 connection record contains a negative value.	7
-13	A comma must separate the Retry, Delay, and Network information fields of an X.21 connection record.	7
-12	The Retry or Delay field of the X.21 connection record contains an invalid character.	7
-11	System does not have enough storage to execute a call request.	7
-10	Not enough storage in partition 1 for X.21 to execute a request.	7
-9	An EDL instruction failed. If \$LOG is active, check the error log record for the X.21 device to find the failing instruction.	7
-2	Text received in conversational mode	
-1	Successful completion	
1	EOT received	
2	DLE EOT received	
3	Reverse interrupt received	
4	Forward abort received	

Return Codes (by Function)

Binary Synchronous Communications (BSC) Return Codes (*continued*)

Return Code	Condition	Notes
5	Remote station not ready (NAK received)	4
6	Remote station busy (WACK received)	4
10	Time-out occurred	1
11	Unrecovered transmission error (BSC error)	1
12	Invalid sequence received	3
13	Invalid multipoint tributary write attempt	2
14	Disregard this block sequence received	1
15	Remote station busy (WACK received)	1
16	Your supervisor does not contain X.21 support.	7
17	The connection type you defined on the BSCLINE statement is not valid for the X.21 Circuit Switched Network.	7
18	The 2080 feature card is incorrectly jumpered for use with the X.21 Circuit Switched Network.	7
19	The X.21 network has been deactivated (DCE CLEAR).	7
20	Wrong length record - long (No COD)	6
21	Wrong length record - short (write only)	2
22	Invalid buffer address	2
23	Buffer length zero	2
24	Undefined line address	2
25	Line not opened by calling task	2
26	Registration or cancellation request processed	7
27	Redirection activated	7
28	Redirection deactivated	7
30	Modem interface error	2
31	Hardware overrun	2
32	Hardware error	5
33	Unexpected ring interrupt	2
34	Invalid interrupt during auto-answer attempt	2
35	Enable or disable DTR error	2
99	Access method error	2

Notes:

1. Retried up to the limit specified in the RETRIES= operand of the BSCLINE statement.
2. Not retried.
3. Retried during write operation only when a wrong ACK is received following an ENQ request after timeout (indicating that no text had been received at the remote station).
4. Returned only during an initial sequence with no retry attempted.
5. Retried only after an unsuccessful start I/O attempt.
6. Retried only during read operations.
7. Returned only if your system contains support for the X.21 Circuit Switched Network.

Return Codes (by Function)

Channel Attach Return Codes

The Channel Attach return codes are placed in the first word of the task control block of the program issuing the CA instruction. The \$CHANUT1 utility also issues the codes to the terminal where it was invoked.

Return Code	Condition
551	Device not started. An attempt was made to open or do I/O to a device that is not started. Start the device with the CASTART instruction or with the utilities before attempting to use it.
552	Stop in progress. An attempt was made to issue a command to a device that is in the process of being stopped. Restart the device before using it.
553	Device in use. A stop command was issued to a device that still has one or more open ports. You must close all ports before stopping the device.
554	Device not found. A start command was issued for a device that does not have a channel attach control block (CACB).
556	Port out of range. A request was made to open a port that is greater than the highest port specified in the channel attach control block (CACB) for the associated device, or a request was made to open a negative port.
557	Port already open. A request was made to open a port that is already open.
558	Read buffer not provided. An address of 0 was specified as the buffer address for the next read operation
559	Read buffer count = 0. A size of zero was specified as the size of the buffer for the next read operation.
560	Buffer not provided. A READ request when an address to 'THISBUF' buffer control block was not provided.
562	Write buffer not provided. A zero was specified as the buffer address for the current write operation.
563	Write buffer count = 0. A zero was specified as the size of the buffer for the current write operation.

Return Codes (by Function)

Channel Attach Return Codes (*continued*)

Return Code	Condition
564	User's CAIOCB not linked to port. The address of the CAIOCB for a read or write operation is inconsistent with the CAIOCB address given when the port was opened.
565	Trace already on. An attempt was made to turn on trace when it is already on.
566	Trace already off. An attempt was made to turn off trace when it is already off.
567	System error - CAPGM terminatng. The CAPGM task error exit module was called because of an error detected in the channel attach code. Channel attach support is unloaded as part of the error cleanup.
568	Port not opened. An attempt was made to perform I/O to a port that is not open. Use the CAOPEN instruction to open a port before attempting to use that port.
569	Device already started. An attempt was made to start a device that is already started.
570	Device in diagnostic mode. An attempt was made to: (1) stop a channel attach device while diagnostics (OLTS) were being run, or (2) to open port 0 while diagnostics were running.
571	CAPGM load error. The channel attach program cannot be loaded properly.
573	Open in progress. An attempt was made to stop a channel attach device while an open request was being processed.
574	Partition number in error. The partition number provided in the buffer control block was not between 1 and 8.

Return Codes (by Function)

Data Formatting Return Codes

The data formatting return codes are placed in the first word of the task control block of the program issuing the instruction.

These return codes are issued by the CONVTB, CONVTD, GETEDIT, and PUTEDIT instructions.

Return Code	Condition
-1	Successful completion
1	No data in field
2	Field omitted
3	Conversion error

Return Codes (by Function)

Disk and Diskette Read/Write Return Codes

Disk and diskette return codes issued by read and write instructions are returned in two places:

- The Event Control Block (ECB) named DSn, where n is the number of the data set being referenced.
- The task code word referred to by 'taskname'.

Several utilities also issue disk and diskette read/write return codes. They issue them in messages that appear on the terminal where the utility was invoked, or on the printer in use.

To obtain further information on a disk or diskette error, print all or part of the contents of the Disk Data Blocks (DDBs) located in the Supervisor. The starting address of the DDBs can be obtained from the linkage editor map of the supervisor. The contents of the DDBs are described in the *Internal Design*. The cycle steal status word and the interrupt status word save areas are of particular value, along with the contents of the word that contains the address of the next DDB in storage.

Note: If an error is encountered during a sequential I/O operation, the relative record number for the next sequential request is not updated. Because the relative record number is not updated, all subsequent I/Os will be in error.

Return Codes (by Function)

Disk and Diskette Read/Write Return Codes (*continued*)

Return Code	Condition
-1	Successful completion
1	I/O error and no device status present (this code may be caused by the I/O area starting at an odd byte address)
2	I/O error trying to read device status
3	I/O error retry count exhausted
4	Read device status I/O instruction error
5	Unrecoverable I/O error
6	Error on issuing I/O instruction for normal I/O
7	A no record found condition occurred, a seek for an alternate sector was performed, and another no record found occurred, for example, no alternate is assigned
8	A system error occurred while processing an I/O request for a 1024-byte sector diskette
9	Device was offline when I/O was requested
10	Record number out of range of data set--may be an end-of-file (data set) condition
11	Data set not open or device marked unusable when I/O was requested
12	DSCB was not OPEN; DDB address = 0
13	If extended deleted record support was requested (\$DCSBFLG bit 3 on), the referenced sector was not formatted at 128 bytes/sector or the request was for more than one 256-byte sector. If extended deleted record support was not requested (\$DCSBFLG bit 3 off), a deleted sector was encountered during I/O.
14	The first sector of the requested record was deleted
15	The second sector of the requested record was deleted
16	The first and second sectors of the requested record were deleted

Note: The actual number of records transferred is in the second word of the TCB.

Return Codes (by Function)

\$DISKUT3 Return Codes

The \$DISKUT3 program places a return code in the first word of a data set control block specified in a DSCB statement.

Return Code	Condition
1	Invalid request code parameter (not 1-6)
2	Volume does not exist (All functions)
4	Insufficient space in library (ALLOCATE)
5	Insufficient space in directory (ALLOCATE)
6	Data set already exists - smaller than the requested allocation
7	Insufficient contiguous space (ALLOCATE)
8	Disallowed data set name, eg. \$\$EDXVOL or \$\$EDXLIB (all functions except OPEN)
9	Data set not found (OPEN, RELEASE, RENAME)
10	New name pointer is zero (RENAME)
11	Disk is busy (ALLOCATE, DELETE, RELEASE, RENAME)
12	I/O error writing to disk (ALLOCATE, DELETE, RELEASE, RENAME)
13	I/O error reading from disk (All functions)
14	Data set name is all blanks (ALLOCATE, RENAME)
15	Invalid size specification (ALLOCATE)
16	Invalid size specification (RELEASE)
17	Mismatched data set type (DELETE, OPEN, RELEASE, RENAME)
18	Data set already exists - larger than the requested allocation
19	SETEOD only valid for data set of type "data"
20	Load of \$DISKUT3 failed (\$RMU only)
21	Tape data sets are not supported
23	Volume not initialized or Basic Exchange Diskette*

* The Basic Exchange Diskette has been opened.

Return Codes (by Function)

EXIO Return Codes

The following codes are issued by the EXIO and EXOPEN instructions, and are returned in word 0 of the TCB. Word 1 of the TCB contains the supervisor instruction address.

Return Code	Condition
-1	Command accepted
1	Device not attached
2	Busy
3	Busy after reset
4	Command reject
5	Intervention required
6	Interface data check
7	Controller busy
8	Channel command not allowed
9	No DDB found
10	Too many DCBs chained
11	No address specified for residual status
12	EXIODEV specified zero bytes for residual status
13	Broken DCB chain (program error)
16	Device already opened
17	Device not opened or already closed

The following codes are issued when an EXIO instruction was completed successfully, but the hardware performing the operation encountered an error. The hardware interrupt condition codes are returned in bits 4 - 7 of the ECB (word 0). If bit 0 is on, then bits 8 - 15 equal the device ID.

Return Code	Condition
0	Controller end
1	Program Controlled Interrupt (PCI)
2	Exception
3	Device end
4	Attention
5	Attention and PCI
6	Attention and exception
7	Attention and device end
8	Not used
9	Not used
10	SE on and too many DCBs chained
11	SE on and no address specified for residual status
12	SE on and EXIODEV specified no bytes for residual status
13	Broken DCB chain
14	ECB to be posted not reset
15	Error in Start Cycle Steal Status (after exception)

Return Codes (by Function)

Floating-Point Return Codes

The floating-point return codes are issued by the FADD, FDIVD, FMULT, and FSUB instructions and are placed in the first word of the task control block of the program issuing the instruction.

Return Code	Condition
-1	Successful completion
1	Floating point overflow
3	Floating point divide check (divide by '0')
5	Floating point underflow

Return Codes (by Function)

Formatted Screen Image Return Codes

These return codes are issued by the \$IMDATA, \$IMOPEN, and \$IMPROT subroutines. They are returned in the second word of the task control block (TCB) of the calling program.

\$IMDATA — Screen Image Unprotected Fields

Return Code	Condition
-1	Successful completion
9	Invalid format in buffer
12	Invalid terminal type

\$IMOPEN — Formatted Screen Image

Return Code	Condition
-1	Successful completion
0	Undefined error encountered
1	Disk I/O error
2	Invalid data set name
3	Data set not found
4	Incorrect header or data set length
5	Input buffer too small
6	Invalid volume name
7	No 3101 image available
8	Data set name longer than eight-bytes

\$IMPROT — Screen Image Protected Fields

Return Code	Condition
-1	Successful completion
9	Invalid format in buffer
10	FTAB truncated due to insufficient buffer size
11	Error in building FTAB from 3101 format; partial FTAB created
12	Invalid terminal type

Return Codes (by Function)

General Purpose Interface Bus (GPIB) Return Codes

The GPIB return codes are issued by the TERMCTRL GPIB instructions and placed in the first word of the task control block of the program containing the instructions. To perform error recovery, retrieve the cycle steal block with the TERMCTRL GPIB,STAT operation and analyze the results.

The \$GPIBUT1 utility also issues the return codes on the terminal where it was loaded.

Return Code	Condition
-1	Successful completion
1	Device not attached
2	busy condition
3	busy after reset
4	command reject
6	Interface data check
256 + ISB	Read exception
512 + ISB	Write exception
1024	Attention received during an operation (may be combined with an exception condition)

Note: "ISB" is the Interrupt Status Byte.

Return Codes (by Function)

Indexed Access Method Return Codes

The following return codes are issued by Indexed Access Method support and by the \$IAMUT1 and \$VERIFY utilities.

For \$IAM support, the codes are placed in the first word of the task control block (TCB) of the program issuing the instruction. For \$IAMUT1 and \$VERIFY, the codes are issued to the terminal where the utility was invoked.

Return Code	Condition
-90	Request cancelled because the request was conditional and a wait on a lock or buffer would be required
-85	Record to be deleted not found
-80	End of data
-79	Warning - File was opened and not closed during the last session, normal processing continues
-75	Warning - File has either not been formatted or the invalid indicator is on in the directory for that file
-58	Record not found
-57	Data set has been loaded
-1	Successful completion
01	Invalid function specified on CALL to \$IAM
07	Link module in use, synchronize use of link module with the program
08	Load error for \$IAM, verify \$IAM exists and enough storage is available to load it
10	Invalid request
12	Data set shut down due to error; see version 2 guide, error recovery
13	A required module is not included in \$IAM
14	Invalid index block found - during processing an incorrect index block type was found, recreate the file
17	IAM is inactive - not enough storage available Use \$IAMUT1 BF command to readjust storage size
22	Address supplied by your program is not a valid IACB
23	Insufficient number of IACBs, use BF command of \$IAMUT1 to allocate more
31	FCB WRITE error during IDEF processing; check system return code
32	Blocksize not multiple of 256
34	Data set is too small
36	Invalid block size during file definition process
37	Invalid record size
38	Invalid index size
39	Record size greater than block size

Return Codes (by Function)

Indexed Access Method Return Codes (*continued*)

Return Code	Condition
40	Invalid number of free records
41	Invalid number of clusters
42	Invalid key size
43	Invalid reserve index value
44	Invalid reserve block value
45	Invalid free pool value
46	Invalid delete threshold value
47	Invalid free block value
48	Invalid number of base records
49	Invalid key position
50	Data set is opened for exclusive use, cannot be opened exclusively
51	Data set already opened in load mode
52	Data set is opened, cannot be opened exclusively
54	\$IAM buffer too small to process a file with this block size Use the BF command of \$IAMUT1 to increase the buffer size
55	Get storage error - FCB
56	READ error - FCB, refer to system return code
60	Out of sequence or duplicate key
61	End of file
62	Duplicate key found in process mode
65	Inconsistent free space parameters RSVIX/RSVBLK or FPOOL/DYN were specified without one or the other
70	No space for insert. Reorganize the file
76	DSOPEN error occurred - The system error field in the OPEN table contains the DSOPEN errors: 21 - DSNAME,VOLUME not found 22 - VOLSERV error 23 - I/O error
77	Record save area not large enough - use \$IAMUT1 BF command to set maximum record size for secondary file processing
78	Attempted to open a secondary file for LOAD, file is not opened independently
80	FCB WRITE error during DELETE processing - see system return code
85	Key field modified by user
90	Internal key save area temporarily in use by another request

Return Codes (by Function)

Indexed Access Method Return Codes (*continued*)

Return Code	Condition
100	READ error - check system return code
101	WRITE error - check system return code
110	WRITE error - data set closed
120	Invalid EXTRACT type
122	File does not contain FCB extension
123	Cannot extract paging statistics. Data paging is not active
150	Not enough storage available for data paging
200	Error occurred while accessing the primary file
201	Request failed because the primary file for this secondary could not be opened; check system return code
210	Request failed because \$DISKUT3 could not be loaded
230	Directory read error from \$IAMDIR
231	\$IAMQCB not found; check sysgen for include of \$IAMQCB
232	Directory open error for \$IAMDIR; verify that directory exists
233	Directory related primary request is a primary entry
234	Directory error - DSNAME,VOL not found in \$IAMDIR
235	Directory resource has not been requested
239	Directory write error; refer to previously displayed message
242	Secondary index is out of sync with primary file. Must rebuild file to get back in sync.
243	Primary file failed to open on secondary open request
244	Primary I/O error
245	Error in opening an auto-update modification request
246	Auto-update processing an INSERT to a secondary failed, auto-update processing continues
247	During auto-update processing a GETSEQ to a secondary failed, auto-update processing continues
248	I/O error on primary file during a secondary request
249	GET UPDATE error occurred trying to update a bad RBN

Note: For return codes 243 through 249, multiple errors may have occurred. Use \$ILOG to display the errors.

Return Codes (by Function)

Job Queue Processor Return Codes

The job queue processor issues these codes in messages that appear on the terminal where the \$JOBQUT or \$SUBMIT utility was loaded.

The \$SUBMITP program also issues return codes in the first word of the task control block of the EVENT= operand you specify in the LOAD instruction.

Return Code	Condition
-1	Job successfully submitted
1	Job queue is full (job not submitted)
2	Invalid data found in job queue data set
3	Disk I/O error while updating job queue data set
4	Unable to load job queue processor

Return Codes (by Function)

LOAD Return Codes

The LOAD instruction return codes are placed in the first word of the task control block of the program issuing the instruction. Several utilities also issue LOAD return codes on the terminal where the utility was invoked.

Note: If you load a program containing an SBIO operation and a sensor I/O error occurs, you will receive an SBIO return code, not a load return code.

Return Code	Condition
-1	Successful completion
61	The transient loader (\$LOADER) is not included in the system
62	In an overlay request, no overlay area exists
63	In an overlay request, the overlay area is in use
64	No space available for the transient loader
65	In an overlay load operation, the number of data sets passed by the LOAD instruction does not equal the number required by the overlay program
66	In an overlay load operation, no parameters were passed to the loaded program
67	A disk(ette) I/O error occurred during the load process
68	Reserved
69	Reserved
70	Not enough main storage available for the program
71	Program not found on the specified volume
72	Disk or diskette I/O error while reading directory
73	Disk or diskette I/O error while reading program header
74	Referenced module is not a program
75	Referenced module is not a data set
76	One of the data sets not found on referenced volume
77	Invalid data set name
78	LOAD instruction did not specify required data set(s)
79	LOAD instruction did not specify required parameters(s)
80	Invalid volume label specified (two or more programs referenced the same volume)
81	Cross partition LOAD requested, support not included at system generation
82	Requested partition number greater than number of partitions in the system
83	Load instruction attempted to access a 1024 bytes/sector diskette without \$IO1024 pre-loaded in storage

Return Codes (by Function)

Message Handler Return Codes

The following return codes are issued by the GETVALUE, READTEXT, QUESTION, and MESSAGE instructions, and are placed in the first word of the task control block of the program containing the instruction. The label of the TCB is the label of the program or task (taskname).

For GETVALUE, READTEXT, and QUESTION, the system issues the codes when you retrieve a prompt message from a disk data set or from processor storage.

Return Code	Condition
300 - 325	Disk message read error (subtract 300 to get the disk read/write return code)
326	Message number out of range
327	Parameter not found
328	No parameters in the instruction
329	Invalid parameter position
330	Invalid parameter type
331	Invalid disk message data set
332	Disk message read error
333	Storage resident module not found
334	Parameter output error

Return Codes (by Function)

Multiple Terminal Manager Return Codes

The following is a summary of the return codes that are returned in a variable you specify when the following functions are called:

- SETPAN
- FILEIO
- FTAB
- SETFMT
- CSEND
- CRECVE
- CHALT
- PSEUDO

Return Code	Condition
-501	Screen data set not found.
-500	Terminal is not an IBM 4978/4979 or 3101; no action has been taken
-5	Destination station is not active
-4	Invalid message length. Either message length is less than 1, or it is greater than the buffer length minus 10. Note that the default length of the Multiple Terminal Manager input buffer is 2048. If the buffer length is insufficient, the message is truncated. For PSEUDO, invalid device name.
-3	Invalid buffer length (buffer length less than 11 for buffer other than the Input Buffer).
-2	FTAB code not link-edited with application. For CRECVE/CSEND, communication disabled. Either Communications Facility is not running, the receiving station cannot be started, or the terminal is not associated with the station. For PSEUDO, the terminal is busy.
-1	Successful completion
1	Warning: for SETPAN, this is an uninitialized panel. Input buffer has been set to unprotected blanks (x'00') and cursor position set to zero. For FTAB, no fields were found. For CRECVE, warning, no message is available. WAIT=0 and message queue is empty.
2	For SETPAN, unprotected data is truncated. For FTAB, the FTAB table is truncated. For SETFMT, data stream is truncated. For CSEND, the destination station does not exist.
3	For SETFMT, no data stream found. For CRECVE, message truncated because larger than buffer size plus 10. Message length reflects size of message actually stored in buffer. For CSEND, no space in Communications Facility message pool.

Return Codes (by Function)

Multiple Terminal Manager Return Codes (*continued*)

Return Code	Condition
4	For CRECVE, receiver's output on hold. For CSEND, sender's input on hold.
5	For CSEND, no space in destination disk queue
6	For CSEND, message longer than disk queue data set
7	Disk queue I/O error
8	For CRECVE, receiver's station is stopped, disk queued message not received. Restart the station.
9	For CSEND, CRECVE, and CHALT, the parameter length is invalid (less than 10 or greater than 32676). Issued only if general bridge request as indicated by non-zero communication code parameter.
10	Communication code error. General bridge.
201	For FILEIO, data set not found. For Communications Facility, no operator requesting that the Communications Facility station be shut down.
202	Volume not found
203	No file table entries are available; all have updates outstanding
204	I/O error reading volume directory
205	I/O error writing volume directory
206	Invalid function request
207	Invalid key operator
208	SEOD record number greater than data set length
Other	Return code from READ/WRITE or Indexed Access Method (FILEIO); disk I/O return code reading program table (PSEUDO)

Return Codes (by Function)

\$PDS Return Codes

The \$PDS program returns the status of an event in the event control block (ECB) specified by the EVENT= parameter on the LOAD instruction.

Return Code	Condition
-1	Successful operation
1	Member not found
2	Member already allocated
3	No space
4	Directory is full
5	Member was not used
7	Record not in member
8	Member control block invalid
9	Space not released
10	Not a data member

Return Codes (by Function)

SBIO (Sensor-Based I/O) Return Codes

The sensor-based I/O return codes are returned in the first word of the task control block of the program issuing the SBIO instruction.

Return Code	Condition
-1	Successful completion
90	Device not attached
91	Device busy or in exclusive use
92	Busy after reset
93	Command reject
94	Invalid request
95	Interface data check
96	Controller busy
97	Analog Input over voltage
98	Analog Input invalid range
100	Analog Input invalid channel
101	Invalid count field
102	Buffer previously full or empty
104	Delayed command reject

Return Codes (by Function)

SDLC Communications Return Codes

The SDLC return codes are returned in the first word of the task control block of the program issuing the instruction.

Return Code	Condition
-351	I/O reset failed
-354	Enable/disable failed
-358	Invalid connection type for SDLC device
-359	Device not jumpered correctly for connection type
-360	You specified an X.21 switched operation, but your system does not contain support for the X.21 Circuit Switched Network.
01	Retry of enable/disable count exceeded
02	Time-out on switched line; hang up
03	Receive data set error; retry count exceeded
04	Send error retry count limit exceeded
05	Invalid cc
06	Disaster ISB; like DCB spec check
07	OIO retry failed
08	Single frame retransmit limit exceeded
09	X.21 connection attempt failed or DCE CLEAR received from the network
10	SNA/SDLC detected an error in your hardware configuration

Return Codes (by Function)

Series/1-to-Series/1 Attachment Return Codes

The Series/1-to-Series/1 return codes are placed in the first word of the task control block of the program issuing TERMCTRL, PRINTEXT and READTEXT instructions. The \$\$S1S1UT1 utility also issues the return codes to the terminal where the utility was loaded.

Return Code	Condition
-1	Successful
1	Device not attached
2	System error (busy condition)
3	System error (busy after reset)
4	System (command reject)
5	Device not ready (not reported for S/1 - S/1)
6	Interface data check
7	Overrun recieved (not reported for S/1 - S/1)
138, 154	An error has occurred that can only be determined by displaying the device cycle steal status word with the TERMCTRL STATUS function and checking the bits to determine the cause of the error
1002	Other system not active
1004	Checksum error detected
1006	Invalid operation code or sequence
1008	Timeout on data transfer
1010	TERMCTRL ABORT issued by responding processor
1012	Device reset (TERMCTRL RESET) issued by the other processor
1014	Microcode load to attachment failed during IPL
1016	Invalid or unsolicited interrupt occurred
1050	TERMCTRL ABORT issued and no operation pending
1052	TERMCTRL IPL attempted by slave processor
1054	Invalid data length

Return Codes (by Function)

Sort/Merge Return Codes

The Sort/Merge return codes indicate successful completion, the issuing of a diagnostic message, or the failure of a system-macro instruction. The codes are returned in the event specified in the EVENT= operand of the LOAD instruction that invoked Sort/Merge.

Return Code	Condition
-1	Successful completion
-2	Warning messages generated during the specification phase; warning message option header statement specified "process"
-3	Warning messages generated during the execution phase; processing continued.
1	Warning message generated during the specification phase; warning message option in header statement specified "terminate"
2	Severe or terminal error occurred during specification phase;
3	Terminal error condition during the execution phase; job terminated
4	Non-zero return code issued after executing a system-macro instruction; job terminated

Return Codes (by Function)

Spool Return Codes

The spool return codes are issued by \$SPOOL. The return codes are sent to the terminal from which you loaded \$SPOOL. If a program loads \$SPOOL, the return codes are sent to the terminal from which you loaded the program.

Return Code	Condition
1	An invalid storage parameter was specified on the \$L \$SPOOL command.
4	Adequate storage for spool control blocks is not available in partition 1.
5	Spooling is already active; only one spool session can run at a time.
7	No valid spool devices specified for the spool session; Spool devices must be printers defined with <code>TERMINAL</code> statement at system generation.
8	Spooling support is not included in the supervisor.
9	The spool data set cannot be restarted. You can: <ul style="list-style-type: none">- Start spooling again from the beginning and rerun the spool job.- Restart spooling with another spool data set.

Return Codes (by Function)

System Return Codes

The system return codes are issued to indicate I/O errors during Indexed Access Method operations. The codes are returned in the FCB.

Return Code	Condition
-1	Successful completion
1	I/O error and no device status present (this code may be caused by the I/O area starting at an odd byte address)
2	I/O error trying to read device status
3	I/O error retry count exhausted
4	Read device status I/O instruction error
5	Unrecoverable I/O error
6	Error on issuing I/O instruction for normal I/O
7	A 'no record found' condition occurred, a seek for an alternate sector was performed, and another 'no record found' occurred, for example, no alternate is assigned
8	A system error occurred while processing an I/O request for a 1024-byte sector diskette
9	Device was 'offline' when I/O was requested
10	Record number out of range of data set--may be an end-of-file (data set) condition
11	Data set not open or device marked unusable when I/O was requested
12	DSCB was not OPEN; DDB address = 0
13	If extended deleted record support was requested (\$DCSBFLG bit 3 on), the referenced sector was not formatted at 128 bytes/sector or the request was for more than one 256-byte sector. If extended deleted record support was not requested (\$DCSBFLG bit 3 off), a deleted sector was encountered during I/O.
14	The first sector of the requested record was deleted
15	The second sector of the requested record was deleted
16	The first and second sectors of the requested record were deleted

Return Codes (by Function)

Systems Network Architecture Return Codes

The return codes for the following Event Driven Executive Systems Network Architecture instructions are listed:

NETCTL
NETGET
NETINIT
NETPUT
NETTERM

The return codes are returned in the first word of the task control block of the program issuing the instruction.

NETCTL Return Codes

The positive return codes from NETCTL TYPE=RECV have bit-significant values to allow for efficient analysis in the Series/1 SNA application. The bit positions have the following meanings:

.... 1 End of transaction received
.... 1. Right-to-send received

The following values are returned in combination with the above bit-significant information:

X'0010' Status message received
X'0020' Message being received from host canceled
X'0030' Session termination request received
X'0050' Request for right-to-send received
X'0060' Host permission to resume sending received
X'0070' Message sent to host rejected

The valid combinations of the values and bit positions are listed in the following decimal return codes.

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

Return Code	Condition
-1	Operation successful
-9	LU is busy with another operation
-10	Session does not exist
-11	Instruction must be issued under program linked to \$NETCMD
-12	Invalid LU number
-13	Invalid request
-14	SNA system error
-15	NETTERM in progress
-16	Session abnormally terminated by host
-17	Status available
-18	Session quiesced
-19	\$SNA never loaded
-20	UNBIND HOLD received
-21	More than two tasks running under this LU. Limit is two tasks.
-22	Session reset. CLEAR and SDT commands received.
-25	Not right-to-send
-26	No status available
1	END BRACKET received
2	CHANGE DIRECTION received
16	LUSTAT received
17	LUSTAT with EB received
18	LUSTAT with CD received
32	CANCEL received
33	CANCEL with EB received
34	CANCEL with CD received
48	SHUTDOWN received
80	SIGNAL received
96	RELQ received
112	Negative response received

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

NETGET Return Codes

The positive return codes from NETGET have bit-significant values to allow for efficient analysis in the Series/1 SNA application. The bit positions have the following meaning:

- 1 Function Management Header received
- 1. End of message received
- 1.. Right-to-send received
- 1... Response to message requested
- 1 End of transaction received
- 1. Start of transaction received

The valid combinations of the bit positions are listed in the following decimal return codes:

Return Code	Condition
-1	Operation successful
-9	LU is busy with another operation
-10	Session does not exist
-11	Instruction must be issued under program linked to \$NETCMD
-12	Invalid LU number
-13	Invalid request
-14	SNA system error
-15	NETTERM in progress
-16	Session abnormally terminated by host
-17	Status available
-19	\$SNA never loaded
-20	UNBIND HOLD received
-21	More than two tasks running under this LU. Limit is two tasks.
-22	Session reset. CLEAR and SDT commands received.
-25	No messages available
-26	Host initiated transaction
1	FMH received
2	End of message received
3	End of message and FMH received
6	End of message and right-to-send received
7	End of message, FMH, and right-to-send received
10	End of message received, response requested received
11	End of message, and FMH received, response requested
14	End of message, and right-to-send received, response requested
15	End of message, FMH, and right-to-send received, response requested
18	End of transaction and end of message received
19	End of transaction, end of message and FMH received
26	End of transaction and end of message received, response requested
27	End of transaction, end of message and FMH received, response requested

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

Return Code	Condition
32	Start of transaction received
33	Start of transaction and FMH received
34	Start of transaction and end of message received
35	Start of transaction, end of message, and FMH received
38	Start of transaction, end of message, and right-to-send received
39	Start of transaction, end of message, FMH, and right-to-send received
42	Start of transaction, end of message, and response requested
43	Start of transaction, end of message, and FMH received, response requested
46	Start of transaction, end of message, and right-to-send received, response requested
47	Start of transaction, end of message, FMH, and right-to-send received, response requested
50	Start and end of transaction, and end of message received
51	Start and end of transaction, end of message and FMH received
58	Start and end of transaction, and end of message received response requested
59	Start and end of transaction, end of message and FMH received, response requested

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

NETINIT Return Codes

The positive return codes from NETINIT have bit-significant values to allow for efficient analysis in the Series/1 SNA application. For a description of the bit-significant values, refer to the Systems Network Architecture.

The following are the decimal return codes that could be returned from a NETINIT operation:

Return Code	Condition
-1	Operation successful
-12	Invalid LU number
-14	SNA system error
-15	NETTERM is progress
-16	Session abnormally terminated by host
-19	\$SNA never loaded
-26	Logical unit already open
-27	No logical unit available
-30	BIND from host rejected
-31	STSN error
-32	No NETTERM HOLD=YES issued
2	Unpresented message from host lost
4	Partially presented message from host lost
17	Message flow to host cold-started, no messages to host lost. Message flow from host cold-started, no messages from host lost
19	Message flow to host cold-started. Message flow from host cold-started, message from host lost
32	Message to host lost
49	Message flow to host cold-started, message to host lost. Message flow from host cold-started no messages from host lost
81	Message flow to host cold-started, message to host possibly lost. Message flow from host cold-started, no messages from host lost

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

NETPUT Return Codes

The positive return codes from NETPUT have bit-significant values to allow for efficient analysis in the Series/1 SNA application. The bit positions have the following meaning:

....1 Host attempted to start a transaction

The valid combinations of the bit positions are listed in the following decimal return codes:

Return Code	Condition
-1	Operation successful
-9	LU is busy with another operation
-10	Session does not exist
-11	Instruction must be issued under program linked to \$NETCMD
-12	Invalid LU number
-13	Invalid request
-14	SNA system error
-15	NETTERM in progress
-16	Session abnormally terminated by host
-17	Status available
-18	Session quiesced
-19	\$SNA never loaded
-20	UNBIND HOLD received
-21	More than two tasks running under this LU. The limit is two tasks.
-22	Session reset. CLEAR and SDT commands received.
-25	Not right-to-send
1	Host attempted to start transaction

Return Codes (by Function)

Systems Network Architecture Return Codes (*continued*)

NETTERM Return Codes

The positive return codes from NETTERM have bit-significant values to allow for efficient analysis in the Series/1 SNA application. The bit positions have the following meaning:

- 1 Message from host rejected during termination
-1. Message to host rejected during termination
-1.. Message to host aborted during termination
- 1... Message from host aborted during termination

The valid combinations of the bit positions are listed in the following decimal return codes:

Return Code	Condition
-1	Operation successful
-10	Session does not exist
-11	Instruction must be issued under program linked to \$NETCMD
-12	Invalid LU number
-14	SNA system error
-15	NETTERM in progress
-16	Session abnormally terminated by host
-19	\$SNA never loaded
-20	UNBIND HOLD received
-25	No UNBIND HOLD received
1	-RSP sent during NETTERM
2	-RSP received during NETTERM
3	-RSP received during NETTERM and -RSP sent during NETTERM
4	CANCEL sent during NETTERM
5	CANCEL sent during NETTERM and -RSP sent
6	CANCEL sent during NETTERM and -RSP received during NETTERM
7	CANCEL sent during NETTERM, -RSP received during NETTERM, and -RSP sent during NETTERM
8	CANCEL received during NETTERM
9	CANCEL received during NETTERM and -RSP sent during NETTERM

Return Codes (by Function)

Systems Network Architecture Extended Error Codes

The Event Driven Executive SNA returns extended error information from the failing function when the ERRCODE parameter is coded on the NETINIT instruction. The return codes from the following functions are listed:

NETBIND
NETCLOSE
NETOPEN
NETRECV
NETSEND
NETUBND
Session termination

Refer to Chapter 4, "Post Codes" on page MC-295 for SNA BIND event post codes. Refer to the disk and diskette read/write return codes for READ/WRITE operations.

NETBIND Return Codes

Return Code	Condition
-9	Session terminated
-13	BIND has not been received yet
-15	Received BIND with invalid parameter
-16	Network aborted
-18	BIND FSM state not reset when checking BIND
-19	BIND FSM state not PEND.ACT when sending response

NETCLOSE Return Codes

Return Code	Condition
-51	Invalid session ID
-301	No buffer available to send TERMSELF
-306	HOLD=YES but no UNBIND HOLD
-307	HOLD=YES but already held

Return Codes (by Function)

Systems Network Architecture Extended Error Codes (*continued*)

NETOPEN Return Codes

Return Code	Condition
-9	Session terminated
-10	Non-sequence procedure error message received
-11	Minus response to INITSELF received
-13	BIND has not been received yet
-14	BIND request received does not compare with user-specified BIND during extended BIND checking
-15	Received BIND with an invalid parameter
-17	Network pending deactivation
-18	Session aborted
-57	No buffer available for INITSELF message
-58	INITSELF message greater than buffer size
-68	SSCP ID in the PUS not the same as user-supplied SSCP ID
-71	UNBIND received
-79	BIND buffer length does not equal user-specified BIND length
-82	No LU-SSCP session (returned when NETOPEN waits on a SSCP-LU session activation and some condition inhibits the activation)
-85	HOLDLU specified but session not closed with HOLD=YES
-86	HOLDLU specifies an invalid LU number
1	UNBIND received

NETRECV Return Codes

Return Code	Condition
-10	CLEAR received
-12	No messages available
1	Successful and message available to be received
2	Message truncated
3	Message truncated and message available to be received

Return Codes (by Function)

Systems Network Architecture Extended Error Codes (*continued*)

NETSEND Return Codes

Return Code	Condition
-9	Session terminated
-10	CLEAR received
-12	Format indicator is on, on this request, and the session is bound no FM headers
-13	Session bound RQN requests only and this request is not RQN
-14	This request carries a BB or EB and not BC
-15	Trying to send a request when the brackets manager is not one of the following states: BETB, PEND.BB, INB PEND.BETB.ECS PEND.BETB.PURGE.S PEND.BETB.S
-16	Trying to send a request when not on a "send" state
-17	QRI=OFF on this request, when sending a set of QRIs
-18	Secondary session quiesced due to a receipt of QEC or SHUTD
-19	Trying to send BC while the chain receive manager is in the INCHAIN state and this request is not a cancel
-20	QC while not in PEND.ACT.QC
-21	CANCEL required. Half session partner in purge state
-22	Trying to send a RQD request that does not carry EC
-24	Data traffic not active
-25	Negative response to SHUTD not allowed
-26	No normal flow response due
-28	Invalid RU code for normal/expedited send request
-29	The form of response is invalid, the response is not RQN, RQE or RQD

Return Codes (by Function)

Systems Network Architecture Extended Error Codes (*continued*)

Return Code	Condition
-31	Response required to an EXR
-32	Positive response to an RQE request not allowed
-33	Invalid data length specified
-34	Maximum RU length exceeded
-35	Response to an outstanding RQD request has not been received
-36	Response to an expedited request has not been received
-37	Negative response to QEC not allowed
-38	Negative response to RELQ not allowed
-39	State error in sending SHUTC while the SHUTD.ECV.FSM not on PEND.ACT.SHUTC
-40	No response due to a normal request with this sequence number
-41	Invalid RU for NRM response
-42	No response due to an expedited request
-43	No response due to an expedited request with this ID
-44	No response due to this expedited RU
-45	Trying to send a DFC command an EB while BSM is in the BETB state
-46	Trying to send an FMD request with no BB while BSM is in the BETB state
-47	Trying to send an FMD request with BB while BSM is in the INB state
-48	Error in trying to send RTR while BSM is in the INB state
-49	Trying to send an FMD request or CANCEL or RTR or any DFC command with EB while BSM is in PEND.BB state
-50	CSI=ON on this request and the session is bound, no alternate code
-51	Secondary logical unit not allowed to send EB
-52	Session bound RQD requests only, and this request is an RQN request
-53	Session bound RQD requests only, and this request is an RQE request with EC
-54	Session bound RQD or RQE requests only, and this request is an RQN request

Return Codes (by Function)

Systems Network Architecture Extended Error Codes (*continued*)

Return Code	Condition
-55	Session bound RQE requests only and this request is not a RQE request
-56	Session bound no brackets, and this request has BB specified
-57	Chain receive manager is in between chains state and this request does not carry BC
-58	Chain receive manager is in between chains state and this is a CANCEL request
-59	The half-duplex manager is in the contention ERP state and this request is not LUSTAT
-60	The half-duplex manager is in the contention ERP state, and this request is LUSTAT with EB and/or CD
-62	Invalid request for the FM profile for which the session was bound
-63	Sending QEC when QECS.FSM not in reset or active state
-64	Invalid request for the TS profile for which the session was bound
-66	CDDCLEAR failed
-67	This change of direction request does not carry EC
-68	Invalid parameter list, does not specify a valid SNA function
-69	Error in trying to send RTR when session is bound without brackets
-70	Negative response to STSN is not allowed
-71	The session was bound single RU chains from the secondary, and this request is not
-72	This function management header request does not carry BC
-73	Response to an outstanding RQR request has not been received
-74	Violation of the session quiesced protocol. QECR FSM or SHUTD FSM are in the P.ACT.QC/P.ACT.SHUTC state
-75	A request without both BB and EB indicators was sent on a duplex session supporting brackets

NETUBND Return Codes

Return Code	Condition
-9	Session terminated
-12	TERMSELF already sent
-13	No buffer available for TERMSELF

Return Codes (by Function)

Systems Network Architecture Extended Error Codes (*continued*)

Session Termination Return Codes

Return Code	Condition
0	Session active
1	BIND not received
2	UNBIND received
3	BIND response not sent
4	CLOSE in progress
5	ACTLU received
6	DACTLU received
7	Buffer count exceeded
8	SDLC failure (hard I/O error)
9	Request disconnect failure
10	Host disconnected or an attempt was made to reconnect the Series/1 while the network was still active
11	ACTPU received
12	DACTPU received
13	Negative response to request disconnect received
14	Network aborted
15	UNBIND HOLD received

Return Codes (by Function)

Tape (READ/WRITE) Return Codes

Tape return codes are issued by the READ, WRITE and CONTROL instructions, and are returned in two places:

1. The Event Control Block (ECB) named DS_n, where n is the number of the data set being referenced
2. The task code word referred to by taskname

Several utilities also issue tape READ/WRITE return codes to the terminal where the utility was invoked.

Return Code	Condition
-1	Successful completion
1	Exception but no status
2	Error reading cycle steal status
3	I/O error; retry count exhausted
4	Error issuing READ CYCLE STEAL STATUS
6	I/O error issuing normal OIO
10	End of data; a tape mark was read
21	Wrong length record
22	Device not ready
23	File protected
24	End of tape
25	Load point
26	Unrecoverable I/O error
27	SL data set not expired
28	Invalid blocksize
29	Offline, in-use, or not open
30	Incorrect device type
31	Close incorrect address
32	Block count error during close
33	Close detected on EOVI

Note: The actual record length or number of records transferred is in the second word of the TCB.

Return Codes (by Function)

Terminal I/O Return Codes

These codes are returned by the PRINTTEXT, READTEXT, and TERMCTRL instructions. The codes differ depending on the type of terminal being accessed. Separate tables show general codes, ACCA, Interprocessor Communications, Virtual Terminal, and 4975 printer return codes. Several utilities also issue terminal I/O return codes to the terminal where the utility was invoked.

Terminal I/O — General

The return codes are placed in the first word of the task control block of the program issuing the instruction.

Return Code	Condition
-1	Successful completion
1	Device not attached
2	System error (busy condition)
3	System error (busy after reset)
4	System error (command reject)
5	Device not ready
6	Interface data check
7	Overrun received
8	Printer power has been switched off and switched back on or a power failure has occurred
>10	A code greater than 10 can indicate multiple errors. To determine the errors, subtract 10 from the code and express the result as an 8-bit binary value. Each bit (numbering from the left) represents an error as follows: Bit 0 - Unused 1 - System error (command reject) 2 - Not used 3 - System error (DCB specification check) 4 - Storage data check 5 - Invalid storage address 6 - Storage protection check 7 - Interface data check

Notes:

1. If the return code is for devices supported by IOS2741 (2741, PROC) and a code greater than 128 is returned, subtract 128; the result then contains status word 1 of the ACCA. Refer to the *IBM Series/1 Communications Features Description*, GA34-0028 to determine the special error condition.
2. If your program receives a return code of 5 while attempting to perform a PRINTTEXT operation on a 4975 printer, the program should retry the operation a maximum of three times.

Return Codes (by Function)

Terminal I/O Return Codes (*continued*)

Terminal I/O — ACCA Return Codes

The return codes are returned in the first word of the task control block of the program issuing the instruction.

Return Code	Condition
-1	Successful completion
1-08	ISB of last operation (I/O complete)
11	Write operation (I/O complete)
12	Read operation (I/O complete)
14,15	Condition code +1 after I/O start or condition code after I/O complete

Terminal I/O — Interprocessor Communications Return Codes

When an Event Driven Executive program is reading transmitted messages, the received end characters (hexadecimal) are stored in the task code word, and indicate the return code. The return codes are as follows for CODTYPE=EBCD/CRSP and CODTYPE=EBCDIC:

CODTYPE=	Return Code	Condition
EBCDIC	FDFE	End of transmission (EOT)
EBCDIC	FEFF	End of record (NL)
EBCDIC	FCFF	End of subrecord (EOSR)
EBCD/CRSP	1F	End of transmission (EOT)
EBCD/CRSP	5B	End of record (NL)
EBCD/CRSP	(none)	End of subrecord (EOSR)

Terminal I/O — Virtual Terminal Communications Return Codes

The return codes are returned in the first word of the task control block of the program issuing the instruction. The meaning of the return code differs if the terminal was receiving or sending.

Return Code	Transmit Condition	Receive Condition
X'8Fnn'	Not applicable	LINE=nn received
X'8Enn'	Not applicable	SKIP=nn received
-2	NA	Line received (no CR)
-1	Successful completion	New line received
1	Not attached	Not attached
5	Disconnect	Disconnect
8	Break	Break

A further description of each of the virtual terminal return codes follows:

Return Codes (by Function)

Terminal I/O Return Codes (*continued*)

LINE=nn (X'8Fnn'): Returned for a READTEXT or GETVALUE instruction if the other program issued an instruction with a LINE= operand. This operand tells the system to perform an I/O operation on a certain line of the page or screen. The return code enables the receiving program to reproduce on an actual terminal the output format intended by the sending program.

SKIP=nn (X'8Enn'): The other program issued an instruction with a SKIP= operand. This operand tells the system to skip a number of lines before performing an I/O operation.

Line Received (-2): Indicates that an instruction (usually READTEXT or GETVALUE) has sent information but has not issued a carriage return to move the cursor to the next line. The information is usually a prompt message.

New Line Received (-1): Indicates transmission of a carriage return at the end of the data. The cursor is moved to a new line. This return code and the Line Received return code help programs to preserve the original format of the data they are transmitting.

Not attached (1): A virtual terminal does not or cannot refer to another virtual terminal.

Disconnect (5): The other virtual terminal program ended because of a PROGSTOP or an operator command.

Break (8): Indicates that both virtual terminal programs are attempting to perform the same type of operation. When one program is reading (READTEXT or GETVALUE), the return code means the other program has stopped sending and is waiting for input. When one program is writing (PRINTTEXT or PRINTNUM), the return code means the other program is also attempting to write.

If you defined only one virtual terminal with SYNC=YES, then that task always receives the break code. If you defined both virtual terminals with SYNC=YES, then the task that last attempted the operation receives the break code.

Terminal I/O — 4975 Printer Return Codes

The return codes are returned in the first word of the task control block of the program issuing the instruction.

Return Code	Condition
301	Invalid TERMCTRL request. Returned for SET function options PDEN, PMODE, and CHARSET. No terminal error exit taken.
302	PRINTTEXT message exceeds line width. Terminal error exit is taken.

Return Codes (by Function)

TP (Host Communication Facility) Return Codes

The return codes are returned in the first word of the task control block of the program issuing the TP instruction.

Return Code	Condition	Module
-1	Successful completion	Supervisor
1	Illegal command sequence	Supervisor
2	TP I/O error	Supervisor
3	TP I/O error on host	HCFCOMM
4	Looping bidding for the line	Supervisor
5	Host acknowledgement to request code was neither ACK0, ACK1, WACK, or a NACK	Supervisor
6	Retry count exhausted - last error was a timeout: the host must be down	Supervisor
7	Looping while reading data from the host	Supervisor
8	The host responded with other than an EOT or an ENQ when an EOT was expected	Supervisor
9	Retry count exhausted - last error was a modem interface check	Supervisor
10	Retry count exhausted - last error was not a timeout, modem check, block check, or overrun	Supervisor
11	Retry count exhausted - last error was a transmit overrun	Supervisor
50	I/O error from last I/O in DSWRITE	DSCLOSE
51	I/O error when writing the last buffer	DSCLOSE
100	Length of DSNAME is zero	HCFCOMM
101	Length of DSNAME exceeds 52	HCFCOMM
102	Invalid length specified for I/O	HCFINIT
200	Data set not on volume specified for controller	HCFINIT
201	Invalid member name specification	DSOPEN
202	Data set in use by another job	DSOPEN
203	Data set already allocated to this task	DSOPEN
204	Data set is not cataloged	DSOPEN
205	Data set resides on multiple volumes	DSOPEN
206	Data set is not on a direct access device	DSOPEN
207	Volume not mounted (archived)	DSOPEN
208	Device not online	DSOPEN
209	Data set does not exist	DSOPEN
211	Record format is not supported	DSOPEN

Return Codes (by Function)

TP (Host Communication Facility) Return Codes (*continued*)

Return Code	Condition	Module
212	Invalid logical record length	DSOPEN
213	Invalid block size	DSOPEN
214	Data set has no extents	DSOPEN
216	Data set organization is partitioned and no member name was specified	DSOPEN
217	Data set organization is sequential and a member name was specified	DSOPEN
218	Error during OS/ OPEN	DSOPEN
219	The specified member was not found	DSOPEN
220	An I/O error occurred during a directory search	DSOPEN
221	Invalid data set organization	DSOPEN
222	Insufficient I/O buffer space available	DSOPEN
300	End of an input data set	DSREAD
301	I/O error during an OS/ READ	DSREAD
302	Input data set is not open	DSREAD
303	A previous error has occurred	DSREAD
400	End of an output data set	DSWRITE
401	I/O error during an OS/ WRITE	DSWRITE
402	Output data set is not open	DSWRITE
403	A previous error has occurred	DSWRITE
404	Partitioned data set is full	DSCLOSE
700	Index, key, and status record added	SET
701	Index exists, key and status added	SET
702	Index and key exist, status replaced	SET
703	Error - Index full	SET
704	Error - Data set full	SET
710	I/O Error	SET
800	Index and key exist	FETCH
801	Index does not exist	FETCH
802	Key does not exist	FETCH
810	I/O error	FETCH
900	Index and/or key released	RELEASE
901	Index does not exist	RELEASE
902	Key does not exist	RELEASE
910	I/O error	RELEASE
1xxx	An error occurred in a subordinate module during SUBMIT. 'xxx' is the code returned by that module.	S7SUBMIT

Return Codes (by Function)

Unmapped Storage Return Codes

The following return codes are issued by the GETSTG, FREESTG, and SWAP instructions. They are returned in the first word of the task control block of the issuing program.

Return Code	Condition
-1	Successful completion
1	A mapped storage entry in storblk already exists (GETSTG) No storage entries exist in storblk (FREESTG) Swap area out of range (SWAP)
2	No mapped storage area available (GETSTG) Swap area not initialized (SWAP)
3	No unmapped storage area available (GETSTG)
4	All unmapped storage entries in the STORBLK are in use (GETSTG)
100	No unmapped storage support in system

Return Codes (by Function)

X.21 Circuit Switched Network Return Codes

The following return codes are issued by the X.21 Circuit Switched Network support.

Return Code	Condition
-32	System is unable to find X.21 support. Re-IPL the system.
-31	Not enough storage available to handle the number of X.21 requests. Use the \$DISKUT2 SS command to allocate more storage for \$X21. You can issue three simultaneous requests for every 256 bytes of storage allocated.
-30	Your supervisor does not contain X.21 support.
-29	System does not have enough storage available to load the X.21 support or the connection record data set, \$\$X21DS, is not on the IPL volume.
-27	Unrecoverable hardware error. If \$LOG is active, check the error log record for the X.21 device for more details.
-26	Hardware error for the 2080 feature card. Invalid interrupt received.
-25	Connection failed
-24	Time expired for the completion of a call request. Call request failed.
-23	You cancelled a call request with an SNADACT or \$C command.
-22	Call request failed due to Public Data Network problems. Call progress signals invalid.
-21	Call request failed due to Public Data Network problems. Call progress signals incomplete.
-20	Call request failed and network would not allow the request to be retried. If \$LOG is active, check the error log record for the X.21 device for more details.
-19	Number of retries exhausted for the call request. If \$LOG is active, check the error log record for the X.21 device for more details.
-18	Hardware error for the 2080 feature card. I/O request could not be completed.
-17	No call request in progress
-16	The Network information field of the X.21 connection record has no plus sign or just a plus sign.
-15	The value in the Retry or Delay field of the X.21 connection record exceeds the maximum value allowed.
-14	The Retry or Delay field of the X.21 connection record contains a negative value.
-13	A comma must separate the Retry, Delay, and Network information fields of an X.21 connection record.
-12	The Retry or Delay field of the X.21 connection record contains an invalid character.
-11	System does not have enough storage to execute a call request.
-10	Not enough storage in partition 1 for X.21 to execute a request.
-9	An EDL instruction failed. If \$LOG is active, check the error log record for the X.21 device to find the failing instruction.
-1	Successful completion
0	You cancelled a call request with an SNADACT or \$C command.
1	Registration or cancellation request processed
2	Redirection activated
3	Redirection deactivated

Return Codes Listed by Number

The following list presents the return codes according to number. When you receive a code, first look it up by the number. Then go down the list until you find the utility, system function or EDL instruction that issued your code.

The “Issued by” column contains the following abbreviations:

- BSC - Binary Synchronous Communications
- Disk READ/WRITE - Disk and Diskette read/write
- GPIB - General Purpose Interface Bus
- IAM - Indexed Access Method
- Job Queue - Job Queue Processor
- Msg Handler - Message Handler
- MTM - Multiple Terminal Manager
- S1S1 - Series/1-to- Series/1 Attachment
- Interprocessor - Terminal I/O Interprocessor Communications
- TP/HCF - Host Communication Facility
- Unmapped Stor - Unmapped Storage
- X.21 - X.21 Circuit Switched Network

Return Codes (Numbered)

Return Code	Issued by	Condition
-1		All -1 codes indicate successful completion
-2	BSC MTM Virtual Terminal Sort/Merge Interprocessor	Text received in conversational mode FTAB not found; module CDMFTAB link edited with application Line received (no CR) Warning message generated during the specification phase. End-of Transmission (EOT)
-3	Sort/Merge Interprocessor	Warning messages generated during the execution phase. Refer to Sort/Merge manual. End-of-Subrecord (EOSR)
-9	BSC NETBIND NETCTL NETGET NETOPEN NETPUT NETSEND NETUBND X.21	An EDL instruction failed. If \$LOG is active, check the error log record for the X.21 device to find the failing instruction Session terminated LU is busy with another operation LU is busy with another operation Session terminated LU is busy with another operation Session terminated Session terminated An EDL instruction failed. If \$LOG is active, check the error log record for the X.21 device to find the failing instruction
-10	BSC NETCTL NETGET NETOPEN NETPUT NETRECV NETSEND NETTERM X.21	Not enough storage in partition 1 for X.21 to execute a request Session does not exist Session does not exist Non-sequence procedure error message received Session does not exist Clear received Clear received Session does not exist Not enough storage in partition 1 for X.21 to execute a request
-11	BSC NETCTL NETGET NETOPEN NETPUT NETTERM X.21	System does not have enough storage to execute a call request Instruction must be issued under program linked to \$NETCMD Instruction must be issued under program linked to \$NETCMD Minus response to INITSELF received Instruction must be issued under program linked to \$NETCMD Instruction must be issued under program linked to \$NETCMD System does not have enough storage to execute a call request

Figure 1 (Part 1 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-12	BSC	The Retry or Delay field of the X.21 connection record contains an invalid character
	NETCTL	Invalid LU number
	NETGET	Invalid LU number
	NETINIT	Invalid LU number
	NETPUT	Invalid LU number
	NETRECV	No messages available
	NETSEND	Format indicator on during this request, and the session is bound on FM header
	NETTERM	Invalid LU number
	NETUBND	TERMSELF already sent
	X.21	The Retry or Delay field of the X.21 connection record contains an invalid character
-13	BSC	A comma must separate the Retry, Delay, and Network information fields of an X.21 connection record
	NETBIND	BIND has not been received yet
	NETCTL	Invalid request
	NETGET	Invalid request
	NETOPEN	BIND has not been received yet
	NETPUT	Invalid request
	NETSEND	Session bound RQN requests only and this request is not RQN
	NETUBND	No buffer available for TERMSELF
X.21	A comma must separate the Retry, Delay, and Network information fields of an X.21 connection record	
-14	BSC	The Retry or Delay field of the X.21 connection record contains a negative value
	NETCTL	SNA system error
	NETGET	SNA system error
	NETINIT	SNA system error
	NETOPEN	BIND request received does not compare with user-specified BIND during extended BIND checking
	NETPUT	SNA system error
	NETSEND	This request carries a BB or EB and not BC
	NETTERM	SNA system error
	X.21	The Retry or Delay field of the X.21 connection record contains a negative value

Figure 1 (Part 2 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-15	BSC	The value in the Retry or Delay field of the X.21 connection record exceeds the maximum value allowed
	NETBIND	Received BIND with invalid parameter
	NETCTL	NETTERM in progress
	NETGET	NETTERM in progress
	NETINIT	NETTERM in progress
	NETOPEN	Received BIND with an invalid parameter
	NETPUT	NETTERM in progress
	NETSEND	Trying to send a request when the brackets manager is not one of the following states: BETB, PEND.BB, INB PEND.BETB.ECS PEND.BETB.PURGE.S PEND.BETB.S
	NETTERM	NETTERM in progress
	X.21	The value in the Retry or Delay field of the X.21 connection record exceeds the maximum value allowed
-16	BSC	The Network information field of the X.21 connection record has no plus sign or just a plus sign
	NETBIND	Network aborted
	NETCTL	Session abnormally terminated by host
	NETGET	Session abnormally terminated by host
	NETINIT	Session abnormally terminated by host
	NETPUT	Session abnormally terminated by host
	NETSEND	Trying to send a request when not in a SEND state
	NETTERM	Session abnormally terminated by host
X.21	The Network information field of the X.21 connection record has no plus sign or just a plus sign	
-17	NETCTL	Status available
	NETGET	Status available
	NETOPEN	Network pending deactivation
	NETPUT	Status available
	NETSEND	QRI=OFF on this request, when sending a set of QRIS
	X.21	No call request in progress

Figure 1 (Part 3 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-18	BSC NETBIND NETCTL NETOPEN NETPUT NETSEND X.21	Hardware error for the 2080 feature card. I/O request could not be completed. BIND FSM state not reset when checking BIND Session quiesced Session aborted Session quiesced Secondary session quiesced due to a receipt of QEC or SHUTD Hardware error for the 2080 feature card. I/O request could not be completed.
-19	BSC NETBIND NETCTL NETGET NETINIT NETPUT NETSEND NETTERM X.21	Number of retries exhausted for the call request. If \$LOG is active, check the error log record for the X.21 device for more details BIND FSM state not PEND.ACT when sending response \$SNA never loaded \$SNA never loaded \$SNA never loaded \$SNA never loaded Trying to send BC while the chain receive manager is in the INCHAIN state and this request is not a cancel \$SNA never loaded Number of retries exhausted for the call request. If \$LOG is active, check the error log record for the X.21 device for more details
-20	BSC NETCTL NETGET NETPUT NETSEND NETTERM X.21	Call request failed and network would not allow the request to be retried. If \$LOG is active, check the error log record for the X.21 device for more details UNBIND-HOLD received UNBIND-HOLD received UNBIND-HOLD received QC while not in PEND.ACT.QC UNBIND-HOLD received Call request failed and network would not allow the request to be retried. If \$LOG is active, check the error log record for the X.21 device for more details
-21	BSC NETCTL NETGET NETPUT NETSEND X.21	Call request failed due to Public Data Network problems. Call progress signals incomplete. More than two tasks already running under this LU More than two tasks already running under this LU More than two tasks already running under this LU CANCEL required; half session partner in purge state Call request failed due to Public Data Network problems. Call progress signals incomplete.

Figure 1 (Part 4 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-22	BSC NETCTL NETGET NETPUT NETSEND X.21	Call request failed due to Public Data Network problems. Call progress signals invalid. Session reset. CLEAR and SDT commands received. Session reset. CLEAR and SDT commands received. Session reset. CLEAR and SDT commands received. Trying to end RQD request that does not carry EC Call request failed due to Public Data Network problems. Call progress signals invalid.
-23	BSC X.21	You cancelled a call request with a \$C command You cancelled a call request with an SNADACT or \$C command
-24	BSC NETSEND X.21	Time expired for the completion of a call request. Call request failed Data traffic not active Time expired for the completion of a call request. Call request failed
-25	BSC NETCTL NETGET NETPUT NETSEND NETTERM X.21	Connection failed Not right-to-send No messages available Not right-to-send Negative response to SHUTD not allowed No UNBIND-HOLD received Connection failed
-26	NETCTL NETGET NETINIT NETSEND X.21	No status available Host initiated transaction Logical unit already open No normal flow response due Hardware error for the 2080 feature card. Invalid interrupt received.
-27	BSC NETINIT X.21	Unrecoverable hardware error. If \$LOG is active, check the error log record for the X.21 device for more details No logical unit available Unrecoverable hardware error. If \$LOG is active, check the error log record for the X.21 device for more details
-28	NETSEND	Invalid RU code for normal/expedited send request
-29	BSC NETSEND X.21	System does not have enough storage available to load the X.21 support or the connection record data set, \$\$X21DS, is not on the IPL volume The form of response is invalid; the response is not RQN, RQE, or RQD System does not have enough storage available to load the X.21 support or the connection record data set, \$\$X21DS, is not on the IPL volume

Figure 1 (Part 5 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-30	BSC NETINIT X.21	Your supervisor does not contain X.21 support BIND from host rejected Your supervisor does not contain X.21 support
-31	BSC NETINIT NETSEND X.21	Not enough storage available to handle the number of X.21 requests. Use the \$DISKUT2 SS command to allocate more storage for \$X21. You can issue three simultaneous requests for every 256 bytes of storage allocated STSN error Response required to an EXR Not enough storage available to handle the number of X.21 requests. Use the \$DISKUT2 SS command to allocate more storage for \$X21. You can issue three simultaneous requests for every 256 bytes of storage allocated
-32	BSC NETINIT NETSEND X.21	System is unable to find X.21 support. Re-IPL the system No NETTERM HOLD=YES issued Positive response to RQE request not allowed System is unable to find X.21 support. Re-IPL the system
-33	NETSEND	Invalid data length specified
-34	NETSEND	Maximum RU length exceeded
-35	NETSEND	Response to an outstanding RQD request has not been received
-36	NETSEND	Response to an outstanding expedited request has not been received
-37	NETSEND	Negative response to QEC not allowed
-38	NETSEND	Negative response to RELQ not allowed
-39	NETSEND	State error in sending SHUTC while the SHUTD.RCV FSM not on PEND.ACT.SHUTC
-40	NETSEND	No response due to a normal request with this sequence number
-41	NETSEND	Invalid RU for NRM response
-42	NETSEND	No response due to an expedited request
-43	NETSEND	No response due to an expedited request with this ID
-44	NETSEND	No response due to this expedited RU
-45	NETSEND	Trying to send DFC command on EB while BSM is in the BETB state
-46	NETSEND	Trying to send FMD request with no BB while BSM is in the BETB state
-47	NETSEND	Trying to send FMD request with BB while BSM in the INB state
-48	NETSEND	Error in trying to send RTR while BSM is in the INB state

Figure 1 (Part 6 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-49	NETSEND	Trying to send FMD request or CANCEL or RTR or any DFC command with EB while BSM is in PEND.BB state
-50	NETSEND	CSI=ON, on this request and the session is bound. no alternate code
-51	NETCLOSE NETSEND	Invalid session ID Secondary logical unit not allowed to send EB
-52	NETSEND	Session bound RQD requests only, and this request is an RQN request
-53	NETSEND	Session bound RQD requests only, and this request is an RQE request with EC
-54	NETSEND	Session bound RQD or RQE requests only and this request is an RQN request
-55	NETSEND	Session bound RQE request only and this request is not an RQE request
-56	NETSEND	Session bound no brackets, and this request has BB specified
-57	IAM NETOPEN NETSEND	Data set has been loaded No buffer available for INITSELF message Chain receive manager is between chains state and this request does not carry BC
-58	IAM NETOPEN NETSEND	Record not found INITSELF message greater than buffer size Chain receive manager is in between chains state and this is a CANCEL request
-59	NETSEND	The half-duplex manager is in the contention ERP state and this request is not LUSTAT
-60	NETSEND	The half-duplex manager is in the contention ERP state, and this request is LUSTAT with EB and/or CD
-62	NETSEND	Invalid request for the FM profile for which the session was bound
-63	NETSEND	Sending QEC when QECS.FSM not in reset or active state
-64	NETSEND	Invalid request for the TS profile for which the session was bound
-66	NETSEND	CDDCLEAR failed
-67	NETSEND	This change of direction request does not carry EC
-68	NETOPEN	SSCP ID in the PUS not the same as user-supplied SSCP ID

Figure 1 (Part 7 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
-68	NETSEND	Invalid parameter list. Does not specify a valid SNA function
-69	NETSEND	Error in trying to send RTR when session is bound without brackets
-70	NETSEND	Negative response to STSN is not allowed
-71	NETOPEN NETSEND	UNBIND recieved The session was bound single RU chains from the secondary, and this request is not
-72	NETSEND	This function management header request does not carry BC
-73	NETSEND	Response to an outstanding RQR request has not been received
-74	NETSEND	Violation of the session quiesced protocol; QECR, FSM or SHUTD FSM are in the P.ACT.QC/P.ACT.SHUT state
-75	IAM NETSEND	Warning - file has either not been formatted or the invalid indicator is on in the directory for that file A request without both BB and EB indicators was sent on a duplex session supporting brackets
-79	IAM NETOPEN	Warning - file was opened and not closed during the last session BIND buffer length does not equal user-specified BIND length
-80	IAM	End of data
-82	NETOPEN	No LU-SSCP session (returned when NETOPEN waits on a SSCP-LU session activation and some condition inhibits the activation)
-85	IAM NETOPEN	Record to be deleted not found HOLDLU specified but session not closed with HOLD=YES
-86	NETOPEN	HOLDLU specified an invalid LU number
-90	IAM	Request cancelled because no wait was requested and IAM would have to wait on a lock or buffer
-301	NETCLOSE	No buffer available to send TERMSELF
-306	NETCLOSE	HOLD=YES but no UNBIND HOLD
-307	NETCLOSE	HOLD=YES but already held
-351	SDLC	I/O reset failed
-354	SDLC	ENABLE/DISABLE failed
-500	MTM	Terminal is not a 4978/4979/3101; no action has been taken
-501	MTM	screen data set not found

Figure 1 (Part 8 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
0	EXIO	Interrupt-controller end
	\$IMOPEN	Undefined error encountered
	Session Termination	Session active
	X.21	You cancelled a call request with an SNADACT or \$C command
1	\$DISKUT3	Invalid request code parm (not 1-5)
	\$IMOPEN	Disk I/O error
	\$PDS	Member not found
	BSC	EOT received
	CONTVD	Invalid data encountered during conversion
	Data Formatting	No data in field
	Disk READ/WRITE	I/O error and no device status present (this may be caused by an odd byte address I/O area)
	EXIO	Device not attached
	EXIO	Program controlled interrupt(PCI)
	FADD	Floating-point overflow
	FDIVD	Floating-point overflow
	Floating Point	Floating-point overflow
	FMULT	Floating-point overflow
	FSUB	Floating-point overflow
	GETEDIT	No data in field
	IAM	Function code not recognized
	MTM	For FTAB, no fields were found, input buffer has been set to unprotected blanks (X'00') and cursor position set to zero Warning: for SETPAN, this is an uninitialized panel
	Job Queue	Job queue is full; job not submitted
	NETCTL	END BRACKET received
	NETGET	FMH received
	NETOPEN	UNBIND received
	NETPUT	Host attempted to start transaction
	NETRECV	Successful and message available to be received
	NETTERM	Negative response received during NETTERM
	PRINTTEXT	Device not attached
	Unmapped Stor	A mapped storage area entry in STORBLK already exists (GETSTG); No storage entries exist in STORBLK (FREESTG)
	Virtual Terminal	Swap area out of range (SWAP)
	Session Termination	Not attached
	Sort/Merge	BIND not received
	SPOOL	Warning message generated during the specification phase
	Tape READ/WRITE	An invalid storage parameter was specified on \$L SPOOL command
	Terminal I/O	Exception but no STATUS
	ACCA	Device not attached
	GPIB	ISB of last operation (I/O complete)
	S1S1	Device not attached
	TP/HCF	Device not attached
	X.21	Invalid command sequence (Supervisor) Registration or cancellation request processed

Figure 1 (Part 9 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
2	\$DISKUT3	Volume does not exist (all functions)
	\$IMOPEN	Invalid data set name
	\$PDS	Member already allocated
	BSC	DLE EOT received
	CONTVD	Field omitted
	Data Formatting	Field omitted
	Disk READ/WRITE	I/O error trying to read device status
	EXIO	Busy
	EXIO	Exception
	GETEDIT	Field omitted
	Job Queue	Invalid data found in the job queue data set
	MTM	For SETPAN, unprotected data is truncated. For FTAB, the FTAB table is truncated. For SETFMT, data stream is truncated.
	NETCTL	CHANGE DIRECTION received
	NETGET	End of message received
	NETINIT	Unpresented message from host lost
	NETRECV	Message truncated
	NETTERM	Negative response received during NETTERM
	PRINTTEXT	System error (busy condition)
	Session Termination	UNBIND received
	Sort/Merge	A severe or terminal error occurred during the specification phase
	Tape READ/WRITE	I/O error trying to read device STATUS
	Terminal I/O	System error (busy condition)
	ACCA	ISB of last operation (I/O complete)
	GPIB	Busy condition
	S1S1	System error (busy condition)
	TP/HCF	TP I/O error
	Unmapped Stor	No mapped storage area available (GETSTG) Swap area not initialized (SWAP)
X.21	Redirection activated	

Figure 1 (Part 10 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
3	\$IMOPEN	Data set not found
	\$PDS	No space
	BSC	Reverse interrupt received
	CONVTB	Conversion error
	CONVTD	Conversion error
	Data Formatting	Conversion error
	Disk READ/WRITE	I/O error retry count exhausted
	EXIO	Busy after reset
	EXIO	Device end
	FDIVD	Floating point divide check (divide by 0)
	Floating Point	Floating point divide check (divide by 0)
	GETEDIT	Conversion error
	Job Queue	Disk I/O error while updating job queue data set
	MTM	No data stream found
	NETGET	End of message and FMH received
	NETRECV	Message truncated and message available to be received
	NETTERM	Response received during NETTERM and negative response sent during NETTERM
	PRINTTEXT	System error (busy after reset)
	PUTEDIT	Conversion error
	Session Termination	BIND response not sent
	Terminal I/O	System error (busy after reset)
	ACCA	ISB of last operation (I/O complete)
	GPIB	Busy after reset
	S1S1	System error (busy after reset)
TP/HCF	TP I/O error on host (HCFCOMM)	
Unmapped Stor	No unmapped storage area available (GETSTG)	
X.21	Redirection deactivated	
4	\$DISKUT3	Insufficient space in library (ALLOCATE)
	\$IMOPEN	Incorrect header or data set length
	\$PDS	Directory is full
	BSC	Forward abort received
	Disk READ/WRITE	Read device status I/O instruction error
	EXIO	Command reject
	EXIO	Attention
	Job Queue	Unable to load job queue processor
	NETINIT	Partially presented message from host lost
	NETTERM	CANCEL sent during NETTERM
	PRINTTEXT	System error (command reject)
	Session Termination	CLOSE in progress
	Sort/Merge	Sort/Merge received a non-zero return code after issuing a system macro instruction and ended the job
	SPOOL	Not enough storage available for spool control blocks in partition 1
	Tape READ/WRITE	Read device status I/O instruction error
	Terminal I/O	System error (command reject)
	ACCA	ISB of last operation (I/O complete)
	GPIB	Command reject
	S1S1	System error (command reject)
	TP/HCF	Looping bidding for the line supervisor
Unmapped Stor	All unmapped storage entries in STORBLK are in use (GETSTG)	

Figure 1 (Part 11 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
5	\$DISKUT3	Insufficient space in directory (ALLOCATE)
	\$IMOPEN	Input buffer too small
	\$PDS	Member was not used
	BSC	Remote station not ready (NAK received)
	Disk READ/WRITE	Unrecoverable I/O error
	EXIO	Intervention required
	EXIO	Attention and PCI
	FADD	Floating-point underflow
	FDIVD	Floating-point underflow
	Floating Point	Floating-point underflow
	FMULT	Floating-point underflow
	FSUB	Floating-point underflow
	NETTERM	CANCEL sent during NETTERM and negative response sent
	PRINTTEXT	Device not ready
	Virtual Terminal	Disconnect
	Session Termination	ACTLU received
	SPOOL	Spooling is already active; only one spool session can run at a time
	Terminal I/O	Device not ready
	ACCA	ISB of last operation (I/O complete)
	TP/HCF	Host acknowledgement to request code was neither ACK0,ACK1,WACK, nor a NACK (Supervisor)
6	\$DISKUT3	Data set already exists - smaller than requested
	\$IMOPEN	Invalid volume name
	BSC	Remote station busy (WACK received)
	DISK R/W	Error issuing I/O instruction for normal I/O
	EXIO	Interface data check
	EXIO	Attention and exception
	NETGET	End of message and right-to-send received
	NETTERM	CANCEL sent during NETTERM and negative response received during NETTERM
	PRINTTEXT	Interface data check
	Session Termination	DACTLU received
	Tape READ/WRITE	Error issuing I/O command
	Terminal I/O	Interface data check
	ACCA	ISB of last operation (I/O complete)
	GPIB	Interface data check
	S1S1	Interface data check
	TP/HCF	Retry count exhausted; last error was a timeout; the host must be down (Supervisor)

Figure 1 (Part 12 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
7	\$DISKUT3	Insufficient contiguous space (ALLOCATE)
	\$IMOPEN	No 3101 image available
	\$PDS	Record not in member
	Disk READ/WRITE	A no-record-found error occurred, an alternate sector seek was done, and a no-record-found reoccurred (no alternate is assigned)
	EXIO	Controller busy
	EXIO	Attention and device end
	IAM	Link module in use
	NETGET	End of message, FMH, and right-to-send received
	NETTERM	CANCEL sent during NETTERM, negative response received during NETTERM, and negative response sent during NETTERM
	PRINTTEXT	Overrun received
	Session Termination	Buffer count exceeded
	Spool	No valid spool devices specified for spool session; spool devices must be printers defined with TERMINAL statements at system generation
	Terminal I/O	Overrun received
	ACCA	ISB of last operation (I/O complete)
S1S1	Overrun received (not reported for S1 - S1)	
TP/HCF	TP looping while reading data from the host (Supervisor)	
8	\$DISKUT3	Disallowed data set name, e.g. \$EDXVOL or \$EDXLIB (all functions)
	\$IMOPEN	Data set name longer than eight bytes
	\$PDS	Member control block invalid
	Disk READ/WRITE	A system error occurred while processing an I/O request for a 1024-byte sector diskette
	EXIO	Channel command not allowed
	EXIO	Not used
	IAM	Load error for \$IAM
	NETTERM	CANCEL received during NETTERM
	Virtual Terminal	Break
	Session Termination	SDLC failure (hard I/O error)
	Spool	Spooling support not included in supervisor
	ACCA	ISB of last operation (I/O complete)
	TP/HCF	The host responded with other than EOT or an ENQ when an EOT was expected (Supervisor)
	9	\$DISKUT3
\$IMDATA		Invalid format in buffer
\$IMPROT		Invalid format in buffer
\$PDS		Space not released
Disk READ/WRITE		Device was offline when I/O was requested
EXIO		No DDB found
EXIO		Not used
NETTERM		CANCEL received during NETTERM and a negative response sent during NETTERM
Session Termination		Request discontact failure
TP/HCF		Retry count exhausted - last error was a modem interface check (Supervisor)

Figure 1 (Part 13 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
10	\$DISKUT3 \$IMPROT \$PDS BSC Disk READ/WRITE EXIO EXIO IAM NETGET Session Termination Tape READ/WRITE TP/HCF	New name pointer is zero (RENAME) FTAB truncated due to insufficient buffer size Not a data member Timeout occurred Record number out of range of data set. This may be an end-of-file (data set) condition Too many DCBS chained (INST) SE on and too many DCBS chained (INTERRUPT) Invalid request End of message received, response requested Host disconnected or an attempt was made to reconnect the Series/1 while the network was still active Tape mark (EOD) Retry count exhausted; last error not a timeout, modem check, block check or overrun (Supervisor)
>10	Terminal I/O	Codes greater than 10 represent possible multiple errors. To determine the errors, subtract 10 from the code and convert result to an 8-bit binary value. Each bit (from left) represents an error as follows: Bit 0 - not used 1 - system error (command reject) 2 - not used 3 - system error (DCB specification check) 4 - storage data check 5 - invalid storage address 6 - storage protection check 7 - interface data check
11	\$DISKUT3 \$IMPROT BSC Disk READ/WRITE EXIO EXIO NETGET Session Termination ACCA TP/HCF	Disk is busy (ALLOCATE, DELETE, RELEASE, RENAME) Error in building FTAB from 3101 format; partial FTAB created Unrecoverable transmission error (BCC error) Data set not open or device marked unusable when I/O requested No address specified for residual status (INST) SE on; no address specified for residual status (INTERRUPT) End of message, and FMH received, response requested ACTPU received I/O complete on a write or control operation Retry count exhausted - last error was a transmit overrun (Supervisor)
12	\$DISKUT3 \$IMDATA \$IMPROT BSC Disk READ/WRITE EXIO EXIO IAM Session Termination ACCA	I/O error writing to disk (ALLOCATE, DELETE, RELEASE, RENAME) Invalid terminal type Invalid terminal type Invalid sequence received DSCB was not open, DDB addr = 0 EXIODEV specified 0 residual status bytes (INST) SE on; EXIODEV specified 0 bytes for residual status (INTERRUPT) Data set shut down due to error DACTPU received Read operation I/O complete

Figure 1 (Part 14 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
13	\$DISKUT3 BSC Disk READ/WRITE EXIO EXIO IAM Session Termination	I/O error reading from disk (all functions) Invalid multipoint tributary write attempt If extended deleted record support requested, (\$DCSBFLG bit 3 on), referenced sector not formatted to 128 bytes per sector, or request was for more than one 256-byte sector. If extended deleted record support not requested (\$DCSBFLG bit 3 off), a deleted sector was encountered Broken DCB chain (program error) (INST) Broken DCB chain (INTERRUPT) A required module is not included in \$IAM; module not included in load module Negative response to request disconnect received
14	\$DISKUT3 BSC Disk READ/WRITE EXIO IAM NETGET Session Termination ACCA	Data set name is all blanks (ALLOCATE, RENAME) Disregard this block sequence received The first sector of requested record deleted ECB to be posted not reset (INTERRUPT) Invalid index block found - index block had incorrect block type End of message, right to send received, response requested Network aborted Condition code +1 after I/O start, or condition code after I/O complete
15	\$DISKUT3 BSC Disk READ/WRITE EXIO NETGET Session Termination ACCA	Invalid size specification (ALLOCATE) Remote station busy (WACK received) The second sector of requested record deleted Error; start cycle steal status after exception (INTERRUPT) End of message, FMH and right to send received, response requested UNBIND-HOLD received Condition code +1 after I/O start, or condition code after I/O complete
16	BSC \$DISKUT3 Disk READ/WRITE EXIO IAM NETCTL	Your supervisor does not contain X.21 support Invalid size specification (RELEASE) The first and second sectors of requested record deleted Device already opened (INST) An error occurred while attempting to open the directory, \$IAMDIR LUSTAT received
17	BSC \$DISKUT3 EXIO IAM NETCTL NETINIT	The connection type you defined on the BSCLINE statement is not valid for the X.21 Circuit Switched Network Mismatched data set type (DELETE, OPEN, RELEASE, RENAME) Device not opened or already closed IAM is inactive - not enough storage available. Use \$IAMUT1 BF command to readjust storage size. LUSTAT with EB received Message flow to or from host cold-started, no messages to or from host lost

Figure 1 (Part 15 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
18	BSC \$DISKUT3 NETCTL NETGET	The 2080 feature card is incorrectly jumpered for use with the X.21 Circuit Switched Network Data set already exists - larger than requested LUSTAT with CD received End of transaction and end of message received
19	BSC \$DISKUT3 NETGET NETINIT	The X.21 network has been deactivated (DCE CLEAR) SETEOD only valid for data sets of type DATA End of transaction, end of message and FMH received Message flow to and from host cold-started, message from host lost
20	\$DISKUT3 BSC	Load of \$DISKUT3 failed for \$RMU Wrong length record - long (no COD)
21	\$DISKUT3 BSC Tape READ/WRITE	Tape data set not supported Wrong length record - short (write only) Wrong record length
22	BSC IAM Tape READ/WRITE	Invalid buffer address Invalid IACB address Drive not ready
23	BSC IAM Tape READ/WRITE	Buffer length zero Get storage error - IACB File protect
24	BSC Tape READ/WRITE	Undefined line address EOT detected
25	BSC Tape READ/WRITE	Line not opened by calling task Load point
26	BSC NETGET Tape READ/WRITE	Registration or cancellation request processed End of transaction and end of message received, response requested Uncorrected I/O error
27	BSC NETGET Tape READ/WRITE	Redirection activated End of transaction, end of message and FMH received, response requested WRITE issued but DS date not expired
28	BSC Tape READ/WRITE	Redirection deactivated Invalid BLKSIZE
29	Tape READ/WRITE	Data set not open
30	BSC Tape READ/WRITE	Modem interface error Incorrect device type
31	BSC IAM Tape READ/WRITE	Hardware overrun FCB WRITE error during IDEF processing; check system return code Incorrect request type on close request

Figure 1 (Part 16 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
32	BSC IAM NETCTL NETGET NETINIT Tape READ/WRITE	Hardware error Blocksize not multiple of 256 CANCEL received Start of transaction received Message to host lost Block count error during close
33	BSC NETCTL NETGET Tape READ/WRITE	Unexpected ring interrupt CANCEL with EB received Start of transaction and FMH received EOV1 label encountered during close
34	BSC IAM NETCTL NETGET	Invalid interrupt during auto-answer Data set too small CANCEL with CD received Start of transaction and end of message received
35	BSC NETGET	Enable or disable DTR error Start of transaction, end of message, and FMH received
36	IAM	Invalid block size during file definition processing
37	IAM	Invalid record size
38	IAM NETGET	Invalid index size Start of transaction, end of message, and right-to-send received
39	IAM NETGET	Record size greater than block size Start of transaction, end of message, FMH and right-to-send received
40	IAM	Invalid number of free records
41	IAM	Invalid number of clusters
42	IAM NETGET	Invalid key size Start of transaction, end of message, and response requested
43	IAM NETGET	Invalid reserve index value Start of transaction, end of message, and FMH received, response requested
44	IAM	Invalid reserve block value
45	IAM	Invalid free pool value
46	IAM NETGET	Invalid delete threshold value Start of transaction, end of message and right-to-send received, response requested
47	IAM NETGET	Invalid free block value Start of transaction, end of message, FMH and right-to-send received, response requested
48	IAM NETCTL	Invalid number of base records SHUTDOWN received

Figure 1 (Part 17 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
49	IAM NETINIT	Invalid key position Message flow to host cold-started, message to host lost; message flow from host cold-started, no messages from host lost
50	IAM NETGET TP/HCF	Data set is opened for exclusive use, cannot be opened exclusively Start and end of transaction, and end of message received I/O error from the last I/O in DSWRITE (DSCLOSE)
51	IAM NETGET TP/HCF	Data set opened in load mode Start and end of transaction, end of message and FMH received I/O error when writing the last buffer (DSCLOSE)
52	IAM	Data set is open; can't be opened exclusively
54	IAM	Invalid block size during PROCESS/LOAD processing
55	IAM	Get storage error - FCB
56	IAM	Read error - FCB
58	NETGET	Start and end of transaction, end of message received, response requested
59	NETGET	Start and end of transaction, end of message and FMH received, response requested
60	IAM	Out of sequence or duplicate key
61	IAM LOAD	End of file The transient loader (\$LOADER) is not included in the system
62	IAM LOAD	Duplicate key found in PROCESS mode In an overlay request, no overlay area exists
63	LOAD	In an overlay request, the overlay area is in use
64	LOAD	No space available for the transient loader
65	IAM LOAD	Inconsistent free space parameters. RSVIX/RSVBLK or FPOOL/DYN specified without one or another In an overlay load operation, the number of data sets passed by the LOAD instruction does not equal the number required by the overlay program
66	LOAD	In an overlay load operation, no parameters were passed to the loaded program
67	LOAD	A disk/diskette I/O error occurred during the load process process
68	LOAD	Reserved
69	LOAD	Reserved
70	IAM LOAD	No space for insert Not enough main storage available for the program
71	LOAD	Program not found on the specified volume
72	LOAD	Disk or diskette I/O error while reading directory
73	LOAD	Disk or diskette I/O error while reading program header

Figure 1 (Part 18 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
74	LOAD	Referenced module is not a program
75	LOAD	Referenced module is not a data set
76	IAM	DSOPEN error occurred. The system error field in the open table contains the DSOPEN errors: 21 - DSNNAME,VOLUME NOT FOUND 22 - VOLSERV ERROR 23 - I/O ERROR
	LOAD	Data set not found on referenced volume
77	IAM	Record save areas not large enough - use \$IAMUT1 BF command to set maximum record size for secondary file processing
	LOAD	Invalid data set name
78	IAM	Attempted to open a secondary file for load
	LOAD	LOAD instruction did not specify required data set
79	LOAD	LOAD instruction did not specify required parameters
80	IAM	FCB write error during delete processing
	LOAD	Invalid volume label specified
	NETCTL	SIGNAL received
81	LOAD	Cross partition LOAD requested, support was not included at system generation
	NETINIT	Message flow to host cold-started; message to host possibly lost Message flow from host cold-started; no message from host lost
82	LOAD	Requested partition number greater than number of partitions in the system
85	IAM	Key field modified by user
90	IAM	Key save area in use
	SBIO	Device not attached
91	SBIO	Device busy or in exclusive use
92	SBIO	Busy after reset
93	SBIO	Command reject
94	SBIO	Invalid request
95	SBIO	Interface data check
96	NETCTL	RELQ received
	SBIO	Controller busy
97	SBIO	Analog input over voltage
98	SBIO	Analog input invalid range
99	BSC	Access method error
100	IAM	Read error; check system return code
	SBIO	Analog input invalid channel
	TP/HCF	Length of DSNNAME is zero (HCFCOMM)
	Unmapped Stor	No unmapped storage support in system
101	IAM	Write error -(check system return code)
	SBIO	Invalid count field
	TP/HCF	Length of DSNNAME exceeds 52 (HCFCOMM)

Figure 1 (Part 19 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
102	SBIO TP/HCF	Buffer previously full or empty Invalid length specified for I/O (HCFINIT)
104	SBIO	Delayed command reject
110	IAM	Write error - data set closed
112	NETCTL	Negative response received
120	IAM	Invalid extract type
122	IAM	Invalid file for type FCBEXT
123	IAM	Data paging not currently active for EXTRACT
138	S1S1	An error has occurred that can only be determined by displaying the device cycle steal status words with the TERMCTRL STATUS function and checking the bits to determine the cause of the error
150	IAM	Not enough storage available for data paging
154	S1S1	See RC code 138 and 1002-1054
200	IAM TP/HCF	Error occurred while accessing the primary file Data set not on volume specified for controller (HCFINIT)
201	IAM MTM TP/HCF	Request failed because primary file for this secondary could not be opened; check system return code Data set not found Invalid member name specification (DSOPEN)
202	IAM MTM TP/HCF	Secondary index is out of sync with primary file Volume not found Data set in use by another job (DSOPEN)
203	IAM MTM TP/HCF	Primary file failed to open on secondary open request No file table entries are available - all have updates outstanding Data set already allocated to this task - (DSOPEN)
204	IAM MTM TP/HCF	Error in opening an auto-update modification request I/O error reading volume directory Data set is not cataloged (DSOPEN)
205	IAM MTM TP/HCF	Auto-update PUTDE to a secondary failed, auto-update processing continues I/O error writing volume directory Data set resides on multiple volumes (DSOPEN)
206	IAM MTM TP/HCF	Auto-update processing an INSERT to a secondary failed; auto-update processing continues Invalid function request Data set not on a direct access device (DSOPEN)
207	IAM MTM TP/HCF	During auto-update processing a GETSEQ to a secondary failed; auto-update processing continues Invalid key operator Volume not mounted (archived) (DSOPEN)
208	IAM MTM TP/HCF	I/O error on primary file during a secondary request SEOD record number greater than data set length Device not inline (DSOPEN)

Figure 1 (Part 20 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
209	IAM TP/HCF	GET UPDATE error occurred trying to update a bad RBN Data set does not exist (DSOPEN)
210	IAM	Request failed because \$DISKUT3 could not be loaded
211	TP/HCF	Record format is not supported (DSOPEN)
212	TP/HCF	Invalid logical record length (DSOPEN)
213	TP/HCF	Invalid block size (DSOPEN)
214	TP/HCF	Data set has no extents (DSOPEN)
216	TP/HCF	Data set organization is partitioned and no member name was specified (DSOPEN)
217	TP/HCF	Data set organization is sequential and a member name was specified (DSOPEN)
218	TP/HCF	Error during OS OPEN (DSOPEN)
219	TP/HCF	The specified member was not found (DSOPEN)
220	TP/HCF	An I/O error occurred during a directory search (DSOPEN)
221	TP/HCF	Invalid data set organization (DSOPEN)
222	TP/HCF	Insufficient I/O buffer space available (DSOPEN)
230	IAM	Directory read error from \$IAMDIR
231	IAM	\$IAMQCB not found; check sysgen for include of \$IAMQCB
232	IAM	Directory open error for \$IAMDIR
233	IAM	Directory related primary request is a primary entry
234	IAM	Directory error - dsname,vol not found in \$IAMDIR
235	IAM	Directory resource has not been requested
239	IAM	Directory write error; refer to previously displayed message
242	IAM	Secondary index is out of sync with primary file. Must rebuild file to get back in sync
244	IAM	Error in opening auto-update on secondary modification request modification request
245	IAM	Auto update PUTDE to a secondary failed, auto-update processing continues.
247	IAM	During auto-update processing a GETSEQ to a secondary failed, auto-update processing continues
248	IAM	I/O error on primary file during a secondary request
249	IAM	GET UPDATE error occurred trying to update a bad RBN
256	GPIB	256 + ISB read exception

Figure 1 (Part 21 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
300	TP/HCF	End of an input data set (DSREAD)
300-325	Msg Handler	Disk message read error (subtract 300 to get the disk read/write return code)
301	TP/HCF	I/O error during an OS READ (DSREAD)
302	TP/HCF	Input data set is not open (DSREAD)
303	TP/HCF	A previous error has occurred (DSREAD)
326	Msg Handler	Message number out of range
327	Msg Handler	Parameter not found
328	Msg Handler	No parameter in the instruction
329	Msg Handler	Invalid parameter position
330	Msg Handler	Invalid parameter type
331	Msg Handler	Invalid disk message data set
332	Msg Handler	Disk message read error
400	TP/HCF	End of an output data set (DSWRITE)
401	TP/HCF	I/O error during an OS WRITE (DSWRITE)
402	TP/HCF	Output data set is not open (DSWRITE)
403	TP/HCF	A previous error has occurred (DSWRITE)
404	TP/HCF	Partitioned data set is full (DSCLOSE)
512	GPIB	512 +ISB write exception
551	Channel Attach	Device not started
552	Channel Attach	Stop in progress
553	Channel Attach	Device in use
554	Channel Attach	Device not found
556	Channel Attach	Port out of range
557	Channel Attach	Port already open
558	Channel Attach	Read buffer not provided
559	Channel Attach	Read buffer count = 0
560	Channel Attach	Buffer not provided
562	Channel Attach	Write buffer not provided
563	Channel Attach	Write buffer count = 0
564	Channel Attach	User's CAIOCB not linked to port
565	Channel Attach	Trace already on
566	Channel Attach	Trace already off
567	Channel Attach	System error;CAPGM terminating
568	Channel Attach	Port not opened
569	Channel Attach	Device already started

Figure 1 (Part 22 of 23). Event Driven Executive Return Codes

Return Codes (Numbered)

Return Code	Issued by	Condition
570	Channel Attach	Device in diagnostic mode
571	Channel Attach	CAPGM load error
573	Channel Attach	Open in progress
574	Channel Attach	Partition number in error
700	TP/HCF	Index, key, and status record added (SET)
701	TP/HCF	Index exists, key and status added (SET)
702	TP/HCF	Index and key exist, status replaced (SET)
703	TP/HCF	Error; index full (SET)
704	TP/HCF	Error; data set full (SET)
710	TP/HCF	I/O error (SET)
800	TP/HCF	Index and key exist (FETCH)
801	TP/HCF	Index does not exist (FETCH)
802	TP/HCF	Key does not exist (FETCH)
810	TP/HCF	I/O error (FETCH)
900	TP/HCF	Index and/or key released (RELEASE)
901	TP/HCF	Index does not exist (RELEASE)
902	TP/HCF	Key does not exist (RELEASE)
910	TP/HCF	I/O error (RELEASE)
999	GPIB	Attention received during an operation (may be combined with an exception condition)
1002	S1S1	Other Series/1 not active
1004	S1S1	Checksum error detected
1006	S1S1	Invalid operation code or sequence
1010	S1S1	TERMCTRL ABORT issued by responding processor
1012	S1S1	Device reset TERMCTRL RESET issued by other processor
1014	S1S1	Microcode load to attachment failed during IPL
1016	S1S1	Invalid or unsolicited interrupt occurred
1024	GPIB	Attention received during an operation (may be combined with an exception condition)
1050	S1S1	TERMCTRL ABORT issued and no operation pending
1052	S1S1	TERMCTRL IPL attempted by responding processor
1054	S1S1	Invalid data length
1XXX	TP/HCF	An error occurred in a subordinate module during SUBMIT; xxx is the code returned by that module.

Figure 1 (Part 23 of 23). Event Driven Executive Return Codes

Chapter 6. Stop Codes

The system uses stop codes to signal error conditions that occur when the system cannot issue error messages or return codes. It is possible to obtain the stop codes only on systems that have programmer consoles.

To obtain the stop codes, set the Mode switch on the Series/1 basic console to Diagnostic mode. If an error condition occurs, the processor will stop. Press the OP REG button on the programmer console to get the stop code. The code is displayed in the console lights in the form "64xx". The "xx" portion of the code indicates the error condition. For a description of how to read the console lights, refer to the *Problem Determination Guide*.

Many times an error condition is shown by a run loop when the console switch is in Normal mode. If a run loop is encountered, set the Mode switch to Diagnostic mode. If it is a stop code type error, the processor will stop. You can then obtain the stop code.

The following section describes the meaning of the values that can appear for the "xx" portion of the stop code. The values are shown in hexadecimal format.

Stop Codes

Stop Code	Module	Error	Action / Response
C4	TIMRINIT	The 7840 timer feature card is not attached or is malfunctioning.	Check the 7840 card to determine if it is attached correctly. If it is attached correctly, the card may be defective.
C5	BSCRINIT	Hardware error for the 2080 feature card or you did not jumper the card to match your software support.	Check the 2080 card to determine if it is jumpered correctly. If it is jumpered correctly, the card may be defective.
CE	\$\$ADUMP	I/O error reading volume label	Bits 0, 1, and 2 of the LSR contain status. Press START to read cycle steal status. Go to stop code E7.
CF	\$\$ADUMP	I/O error reading volume label	Press START to retry operation. If condition continues, recreate the dump diskette using \$DASDI.
D3	\$OVLMGRO	I/O error while reading initialization overlay routines.	Re-IPL. If you receive the stop code again, regenerate the nucleus.
DA	INIT4980	<ol style="list-style-type: none"> 1. Read error for random access memory load data set. 2. Error during load of 4980 terminal random access memory. 	<ol style="list-style-type: none"> 1. Recopy the image store (\$4980ISO), the control store (\$4980CS0), and the microcode (\$4980R01) to the IPL volume. 2. Hardware error.
DB	INITMFA	I/O error writing random access memory record to the MFA.	Bits 0, 1, and 2 of level status register (LSR) contain I/O condition code or interrupt condition code. The initialization module INITMFA terminates.
DC	INITMFA	Read error on MFA random access memory load data set.	The initialization module INITMFA terminates. Recopy \$MFARAM to IPL volume.
DD	INITMFA	Open error on the multifunction adapter (MFA) random access memory load data set.	Ensure that \$MFARAM resides on the IPL volume. Use the \$INITDSK commands, VD (to verify the device), and VV (to verify the volume).
DE	INITMFA	Error reading correct random access memory load data set.	Recopy \$MFARAM to IPL volume.
DF	INITMFA	Error finding correct random access memory data set.	Ensure that \$MFARAM resides on IPL volume.
E0	\$\$ADUMP	I/O error reading stand alone dump program off diskette.	Recreate the stand alone dump diskette using \$DASDI

Figure 2 (Part 1 of 3). Stop codes

Stop Codes

Stop Code	Module	Error	Action / Response
E1	\$\$ADUMP	Unexpected class interrupt	None. Re-IPL.
E2	\$\$ADUMP	I/O error writing control blocks for stand alone dump program	Press START to retry operation. If condition continues, recreate the dump diskette using \$DASDI.
E3	\$\$ADUMP	I/O error writing control blocks for stand alone dump program	Bits 0, 1, and 2 of the LSR contain status. Press START to read cycle steal status. Go to stop code E7.
E4	\$\$ADUMP	I/O error on Seek/Recalibrate for stand alone dump program	(see response for code E3)
E5	\$\$ADUMP	I/O failed on storage dump.	(see response for code E2)
E6	\$\$ADUMP	I/O error interrupt writing storage	(see response for code E3)
E7	\$\$ADUMP	Read cycle steal status after error.	Console lights show contents of cycle steal status word 2. Dump program terminates.
E8	\$\$ADUMP	Stand alone dump has completed successfully.	None
E9	\$TRAP	\$TRAP successful	Re-IPL.
EA	\$TRAP	Error reading nucleus	Move nucleus to another volume. Rerun \$TRAP.
EB	INIT4978	Processor storage parity error during load of 4978 terminal random access memory.	Hardware error.
EC	INITADAP	Too many loads of random access memory have been performed for this adapter card (limit is 99).	Severe hardware problem. Adapter card is unusable.
ED	INITADAP	I/O error while writing random access memory record to the adapter.	Bits 0, 1, and 2 of the level status register (LSR) contain I/O condition code or interrupt condition code. Examine them to determine cause of the problem.
EE	INITADAP	Read error for random access memory load data set.	Verify the disk in use with the \$INITDSK utility. Then recopy the random access memory load data set to the IPL volume.
EF	INITADAP	Open error on the random access memory load data set.	Ensure that the random access memory load data set is in on the IPL volume.
F0	SEGINIT	Not enough physical storage for supervisor code.	Generate a smaller system (with fewer I/O devices and/or fewer software features).

Figure 2 (Part 2 of 3). Stop codes

Stop Codes

Stop Code	Module	Error	Action / Response
F1	XPSINIT	DDB chain or IPL device DDB undefined.	Use \$IOTEST to find which disk was not included in the system. Regenerate the system, including the device.
F2	XPSINIT	Disk(ette) device type is undefined.	Use \$IOTEST to find which device was incorrectly defined. Regenerate the system, correcting the device definition.
F3	XPSINIT	I/O error on seek/recalibrate (hardware error).	Copy nucleus to another volume. Re-IPL.
F4	XPSINIT	Seek/recalibrate interrupt was bad.	Copy nucleus to another volume. Re-IPL.
F5	XPSINIT	I/O error reading IPL text from disk(ette). Device type incorrectly defined.	Use \$IOTEST to find which device is incorrectly defined. Regenerate the system; then re-IPL.
F6	XPSINIT	Interrupt on read of IPL text from disk(ette) was bad. Device type incorrectly defined at IPL.	Use \$IOTEST to find which device is incorrectly defined. Re-IPL, writing text to another volume.
F7	XPSINIT	XPS header record was not found.	Regenerate the nucleus. Re-IPL.
F8	XPSINIT	Device type in XPSINIT IPL text is undefined.	Copy the nucleus to another volume. Re-IPL.
F9	XPSINIT	I/O failed during load of XPS segments.	Copy the nucleus to another volume. Re-IPL. If you receive the stop code again, regenerate the nucleus and re-IPL.
FB	EDXSVCX	IA buffer overflow	Re-IPL. Consider a system generation to increase the IABUF parameter on the SYSTEM statement. (default is 20).
FC	EDXSVCX	Overflow; no free XPSTK entries	Re-IPL. Increase the XPSTK parameter of the SYSTEM statement during system generation (default is 20).
FD	EDXSVCX	Invalid TCBINDX	Re-IPL.
FE	EDXSVCX	Invalid SVC request	Press the start button.

Figure 2 (Part 3 of 3). Stop codes

Appendix A. Program Checks and Processor Status Word (PSW)

Program check messages provide status information on errors that occur during processing of programs. These messages are displayed in the terminal defined as `$$SYSLOG`.

The supervisor issues two types of program check messages: system program check and application program check.

The information in this appendix applies to programs written in the Event Driven Language.

How to Interpret the Program Check Message

The program check message can be in one of the following two formats:

- Format 1 is the standard type of message that the supervisor issues when you have not specified a task error exit in your program (`ERRXIT=` operand is not coded in the `PROGRAM` or `TASK` statement).
- Format 2 is displayed when you specify `$$EDXIT` as the task error exit for your program (`ERRXIT=$$EDXIT` is coded in the `PROGRAM` or `TASK` statement).

Format 1: Standard Program Check Messages

For programs with no task error exit specified, the supervisor issues two kinds of program check messages: application program checks, and system program checks.

Program Checks and Processor Status Word (PSW)

How to Interpret the Program Check Message (*continued*)

The following is an example of the standard application program check message:

```
PROGRAM CHECK:
PLP  TCB  PSW  IAR  AKR  LSR  R0   R1   R2   R3   R4   R5   R6   R7
3A00 0120 8002 2AD6 0110 80D0 0064 3B0A 3B20 3A37 3A34 015C 00B8 0000
```

The next example shows the system program check message:

```
SYSTEM PGM CHECK:
PSW  IAR  AKR  LSR  R0   R1   R2   R3   R4   R5   R6   R7
8002 2AD6 0110 80D0 0064 3B0A 3B20 3A37 3A34 015C 00B8 0000
```

The headings displayed in the message have the following meanings.

PLP The address in storage of the program load point. This is the address at which the program was loaded for execution and represents the first word of your program listing.

For a system program check message, this field is omitted because the failing instruction is within the supervisor.

TCB The address in storage of the active task control block (TCB).

For a system program check message, this field is omitted because the failing instruction is within the supervisor.

PSW The value of the processor status word (PSW) when the program check occurred. Refer to the section "Processor Status Word" on page MC-381 to determine the meaning of this value.

IAR The contents of the instruction address register (IAR) at the time of the error. The value shown is the address of the machine instruction currently executing.

AKR The value of the address key register (AKR) at the time of the error. This last 3-hexadecimal digits indicate in which address space operand 1 and operand 2 reside and in which address space the IAR is executing.

LSR The value of the level status register (LSR) when the error occurred. This information indicates the status of the last assembler instruction executed.

The next portion of the program check message displays the contents of the hardware registers R0–R7. If the failing program is written in a language other than EDL, refer to the user's guide for that language to determine the register usage.

R0 Because the supervisor uses this register as a work register, the contents are usually not significant to the failing program.

How to Interpret the Program Check Message (*continued*)

- R1 The address in storage of the failing EDL instruction.
- R2 The address in storage of the active task control block (TCB).
- R3 The address in storage of EDL operand 1 of the failing instruction.
- R4 The address in storage of EDL operand 2 of the failing instruction (if applicable).
- R5 Byte 2 shows the EDL operation code of the failing instruction.
- R6 Because the supervisor uses this register as a work register, the contents are usually not significant to the failing program. However, if the value in R6 is equal to twice the value of the operation code (second byte of R5 multiplied by 2), this indicates that the processor was emulating EDL code.
- R7 Because the supervisor uses this register as a work register, the contents are usually not significant to the failing program.

After reviewing the information shown in the program check message, you must analyze the contents displayed for the *processor status word* (PSW).

The processor status word is a 16-bit register the system uses to save error status. By looking at the processor status word, you can determine whether the error is hardware or software related. The next section explains how to interpret the processor status word.

Processor Status Word

In the standard program check messages, the processor status word (PSW) consists of 4 hexadecimal digits. Each hexadecimal digit represents the sum of 4 binary bits. Starting from left to right, the value of each bit (when set) is 8, 4, 2, and 1. To interpret the bit settings, you must convert each hexadecimal digit to binary. For example, if the PSW showed the value X'8002', the binary representation and the bit positions would be as shown below:

Hex value	Binary value	PSW bits
8	1000	0 - 3
0	0000	4 - 7
0	0000	8 - 11
2	0010	12 - 15

Figure 3. Sample processor status word bit settings

In the previous example, note that bits 0 and 14 are set. These bit settings are the same as X'8002'.

After you convert the value to binary and identify which bit positions are set, refer to "Processor Status Word Bit Descriptions" on page MC-382 for an explanation of what each bit indicates. Remember that bit 0 is the leftmost bit in the 16-bit string.

Program Checks and Processor Status Word (PSW)

How to Interpret the Program Check Message (*continued*)

Processor Status Word Bit Descriptions

Figure 4 shows the PSW bits and their general assignment for the different processors. An explanation of the bit settings follows Figure 4.

Refer to the specific processor description manual for details on class interrupts, I/O interrupts, and the basic instruction set (including indicator settings and possible exceptions conditions).

Bit	Processor type 495x 2 3 4 5 6	Condition	Class interrupt
0	X X X X X	Specification check	Program check
1	X X X X X	Invalid storage address	Program check
2	X X X X X	Privilege violate	Program check
3	X X X X	Protect check	Program check
4	X X X X X	Invalid function	Soft-exception
5	X X X	Floating-point exception	Soft-exception
6	X X X X X	Stack exception	Soft-exception
7		Not used	
8	X X X X X	Storage parity check	Machine check
9		Not used	
10	X X X X X	Processor control check	Machine check
11	X X X X X	I/O check	Machine check
12	X X X X X	Sequence indicator	None
13	X X X X X	Auto IPL	None
14	X X X X	Translator enabled	None
15	X X X X X	Power/thermal warning	Power/thermal

Figure 4. Processor status word bit assignments

Processor Status Word Bit Descriptions

The information in the processor status word falls into three categories:

- Software problems — bits 0–6
- Hardware problems — bits 8, 10, or 11
- Processor status — bits 12–15

Each of the bits is described below.

- Bit 0, Specification Check.

Set to 1 if (a) the storage address violates the boundary requirements of the specified data type, or (b) the effective (computed) address is odd when attempting to execute a floating-point instruction and the floating-point feature is not installed.

- Bit 1, Invalid Storage Address. Set to 1 when an attempt is made to access a storage address outside the storage size of the system.

How to Interpret the Program Check Message (*continued*)

- **Bit 2, Privilege Violate.** Set to 1 if a program in problem state attempts to issue a privileged instruction. The processor can run in either supervisor or problem state. Some assembler instructions can be used only while in supervisor state. If an assembler program in problem state attempts to issue a privileged instruction, the privilege violate condition occurs. This error normally never occurs in an EDL program.
- **Bit 3, Protect Check.** Set to 1 if a program attempts to access protected storage. The processor can control access to areas in storage by using a storage protect feature. If a program attempts to address any part of the protected storage, the protect check indicator is set. This error normally never occurs in an EDL program.
- **Bit 4, Invalid Function.** Set to 1 if any of the following conditions occur:
 - Attempted execution of an illegal operation code or function combination.
 - The processor attempts to execute an instruction associated with a feature that is not installed.
- **Bit 5, Floating-Point Exception.** Set to 1 when an exception condition is detected by the optional floating-point processor. Floating-point hardware sets this bit to indicate underflow, overflow, and divide check exceptions. An EDL program can detect these exceptions by the return code from floating-point instruction. No program check message is issued when this exception occurs.
- **Bit 6, Stack Exception.** Set to 1 when an attempt has been made to pop an operand from an empty processor storage stack or push an operand into a full processor storage stack. A stack exception also occurs when the stack cannot contain the number of words to be stored by an assembler Store Multiple (STM) instruction. This error normally never occurs in an EDL program.
- **Bit 8, Storage Parity.** Set to 1 when the hardware detects a parity error on data being read out of storage by the processor.
- **Bit 10, Processor Control Check.** Set to 1 if no levels are active but execution continues.
- **Bit 11, I/O Check.** Set to 1 when a hardware error has occurred on the I/O interface that may prevent further communication with any I/O device.
- **Bit 12, Sequence Indicator.** Set to 1 to reflect the last I/O interface sequence to occur. This indicator is used in conjunction with I/O check (bit 11).
- **Bit 13, Auto IPL.** Set to 1 by the hardware when an automatic IPL occurs.
- **Bit 14, Translator Enabled.** Set to 1 when the Storage Address Relocation Translator Feature is installed and enabled.

Program Checks and Processor Status Word (PSW)

How to Interpret the Program Check Message (*continued*)

- Bit 15, Power Warning and Thermal Warning. Set to 1 when these conditions occur (refer to the appropriate processor manual for a description of a power/thermal warning class interrupt).

Format 2: Program Check Message from \$\$EDXIT

The \$\$EDXIT task error exit causes the supervisor to issue a program check message with descriptive headings, which provides more information than the standard program check messages. \$\$EDXIT also provides an analysis of the processor status word.

When a program check occurs, the program check message is directed to \$\$SYSLOG and \$\$SYSPRTR.

The following is an example of a program check message issued by \$\$EDXIT. An explanation of each item in the sample output follows the example.

```
*****
* WARNING!! AN EXCEPTION HAS OCCURRED!! *
*****

PROGRAM NAME           = PCHECK      PSW = 8002
PROGRAM VOLUME         = MYVOL       IAR = 2A06
PROGRAM LOAD POINT     = 0000        AKR = 0110
ADDRESS OF ACTIVE TCB = 0120        LSR = 80D0
ADDRESS OF CCB         = 0F5E        R0 (WORK REGISTER) = 0064
NUMBER OF DATA SETS   = 0           R1 (EDL INSTR ADDR) = 010A
NUMBER OF OVERLAYS     = 0           R2 (EDL TCB ADDR)   = 0120
$TCBADS                = 0001        R3 (EDL OP1 ADDR)   = 0037
ADDRESS OF FAILURE     =              R4 (EDL OP2 ADDR)   = 0034
  (REL. TO PGM LOAD PT) = 010A        R5 (EDL COMMAND)    = 015C
DUMP OF FAIL ADDRESS   =              R6 (WORK REGISTER) = 00B8
  010A: 015C 0000 0034 8332          R7 (WORK REGISTER) = 0000
$TCBC0 = -1 DEC; FFFF HEX          #1 = 0037
$TCBC02 = 0 DEC; 0000 HEX          #2 = 0000
PSW ANALYSIS:

SPECIFICATION CHECK
TRANSLATOR ENABLED
```

The fields of the \$\$EDXIT program check message are described below.

- **PROGRAM NAME** identifies the name of the failing application program. In this example, the program PCHECK failed.
- **PSW** indicates the value of the *processor status word* when the error occurred. \$\$EDXIT interprets this value and displays its meaning as shown in the **PSW ANALYSIS** field.

A detailed description of the processor status word and the associated bits are presented in the section “Processor Status Word Bit Descriptions” on page MC-382.

How to Interpret the Program Check Message (*continued*)

- **VOLUME NAME** identifies the name of the volume from which the failing application program was loaded. In this example, the name of the volume is MYVOL.
- **IAR** (instruction address register) contains the address of the currently executing assembler instruction.
- **PROGRAM LOAD POINT** contains the address at which the program was loaded for execution. The address represents the first word of your program listing.
- **AKR** (address key register) shows the address space of the currently executing assembler instruction and its operands.
- **ADDRESS OF THE ACTIVE TCB** contains the address of the active task control block (TCB).
- **LSR** (level status register) indicates the status of the last assembler instruction executed.
- **ADDRESS OF CCB** contains the address of the terminal control block (CCB) assigned to the failing program.
- **R0** contains the contents of hardware register 0 when the error occurred. Because the supervisor uses this register as a work register, the contents are usually not significant when you analyze the failing program.
- **NUMBER OF DATA SETS** shows the number of data sets that were open for use by the failing program.
- **R1** contains the address of the failing EDL instruction.
- **NUMBER OF OVERLAYS** indicates the number of overlay programs specified on the PGMS= operand of the PROGRAM statement.
- **R2** contains the address of the active task control block.
- **\$TCBADS** contains the target task address space. The value of this field plus 1 indicates the partition number in which the program was running.
- **R3** contains the address of operand 1 for the failing EDL instruction.
- **ADDRESS OF FAILURE** contains the address of the failing EDL instruction. This is the address shown in the compiler listing. This is also the address shown in the R1 field. In this example, the failing EDL instruction is at address X'010A'.
- **R4** contains the address of operand 2 for the failing EDL instruction.
- **R5** contains the EDL operation code of the instruction that was executing when the failure occurred.

Program Checks and Processor Status Word (PSW)

How to Interpret the Program Check Message (*continued*)

- **DUMP OF FAIL ADDRESS** shows the location and content of the instruction that was executing when the failure occurred. The information at this address also appears in the compiler listing.
- **R6** contains the contents of general purpose register 6 when the error occurred. Because the supervisor uses this register as a work register, the contents are usually not significant when you analyze the failing program.
- **R7** contains the contents of general purpose register 7 when the error occurred. Because the supervisor uses this register as a work register, the contents are usually not significant when you analyze the failing program.
- **\$TCBCO** shows the value in the first word of the failing program's task control block (TCB). The value is displayed in decimal and followed by the hexadecimal equivalent.
- **#1** shows the contents of index register 1 when the failure occurred. In this example, #1 contains the value X'0037'.
- **\$TCBCO2** shows the value in the second word of the failing program's task control block (TCB). The value is displayed in decimal and followed by the hexadecimal equivalent.
- **#2** shows the contents of index register 2 when the failure occurred.
- **PSW ANALYSIS** explains the meanings of the bit settings in the processor status word (PSW). The hexadecimal format of the processor status word is shown in the PSW field. This information indicates the type of error that occurred.

Glossary of Terms and Abbreviations

This glossary defines terms and abbreviations used in the Series/1 Event Driven Executive software publications. All software and hardware terms pertain to EDX. This glossary also serves as a supplement to the *IBM Data Processing Glossary*, GC20-1699.

\$\$SYSLOGA, \$SYSLOGB. The name of the alternate system logging device. This device is optional but, if defined, should be a terminal with keyboard capability, not just a printer.

\$\$SYSLOG. The name of the system logging device or operator station; must be defined for every system. It should be a terminal with keyboard capability, not just a printer.

\$\$SYSPRTR. The name of the system printer.

abend. Abnormal end-of-task. Termination of a task prior to its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

ACCA. See asynchronous communications control adapter.

address key. Identifies a set of Series/1 segmentation registers and represents an address space. It is one less than the partition number.

address space. The logical storage identified by an address key. An address space is the storage for a partition.

application program manager. The component of the Multiple Terminal Manager that provides the program management facilities required to process user requests. It controls the contents of a program area and the execution of programs within the area.

application program stub. A collection of subroutines that are appended to a program by the linkage editor to provide the link from the application program to the Multiple Terminal Manager facilities.

asynchronous communications control adapter. An ASCII terminal attached via #1610, #2091 with #2092, or #2095 with #2096 adapters.

attention key. The key on the display terminal keyboard that, if pressed, tells the operating system that you are entering a command.

attention list. A series of pairs of 1 to 8 byte EBCDIC strings and addresses pointing to EDL instructions. When the attention key is pressed on the terminal, the operator can enter one of the strings to cause the associated EDL instructions to be executed.

backup. A copy of data to be used in the event the original data is lost or damaged.

base record slots. Space in an indexed file that is reserved for based records to be placed.

base records. Records are placed into an indexed file while in load mode or inserted in process mode with a new high key.

basic exchange format. A standard format for exchanging data on diskettes between systems or devices.

binary synchronous device data block (BSCDDB). A control block that provides the information to control one Series/1 Binary Synchronous Adapter. It determines the line characteristics and provides dedicated storage for that line.

Glossary of Terms and Abbreviations

block. (1) See data block or index block. (2) In the Indexed Method, the unit of space used by the access method to contain indexes and data.

block mode. The transmission mode in which the 3101 Display Station transmits a data stream, which has been edited and stored, when the SEND key is pressed.

BSCAM. See binary synchronous communications access method.

binary synchronous communications access method. A form of binary synchronous I/O control used by the Series/1 to perform data communications between local or remote stations.

BSCDDB. See binary synchronous device data block.

buffer. An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. See input buffer and output buffer.

bypass label processing. Access of a tape without any label processing support.

CCB. See terminal control block.

central buffer. The buffer used by the Indexed Access Method for all transfers of information between main storage and indexed files.

character image. An alphabetic, numeric, or special character defined for an IBM 4978 Display Station. Each character image is defined by a dot matrix that is coded into eight bytes.

character image table. An area containing the 256 character images that can be defined for an IBM 4978 Display Station. Each character image is coded into eight bytes, the entire table of codes requiring 2048 bytes of storage.

character mode. The transmission mode in which the 3101 Display Station immediately sends a character when a keyboard key is pressed.

cluster. In an indexed file, a group of data blocks that is pointed to from the same primary-level index block, and includes the primary-level index block. The data records and blocks

contained in a cluster are logically contiguous, but are not necessarily physically contiguous.

COD (change of direction). A character used with ACCA terminal to indicate a reverse in the direction of data movement.

cold start. Starting the spool facility by erasing any spooled jobs remaining in the spool data set from any previous spool session.

command. A character string from a source external to the system that represents a request for action by the system.

common area. A user-defined data area that is mapped into the partitions specified on the SYSTEM definition statement. It can

be used to contain control blocks or data that will be accessed by more than one program.

completion code. An indicator that reflects the status of the execution of a program. The completion code is displayed or printed on the program's output device.

constant. A value or address that remains unchanged throughout program execution.

controller. A device that has the capability of configuring the GPIB bus by designating which devices are active, which devices are listeners, and which device is the talker. In Series/1 GPIB implementation, the Series/1 is always the controller.

conversion. See update.

control station. In BSCAM communications, the station that supervises a multipoint connection, and performs polling and selection of its tributary stations. The status of control station is assigned to a BSC line during system generation.

cross-partition service. A function that accesses data in two partitions.

cross-partition supervisor. A supervisor in which one or more supervisor modules reside outside of partition 1 (address space 0).

data block. In an indexed file, an area that contains control information and data records. These blocks are a multiple of 256 bytes.

data record. In an indexed file, the records containing customer data.

data set. A group of records within a volume pointed to by a directory member entry in the directory for the volume.

data set control block (DSCB). A control block that provides the information required to access a data set, volume or directory using READ and WRITE.

data set shut down. An indexed data set that has been marked (in main storage only) as unusable due to an error.

DCE. See directory control entry.

device data block (DDB). A control block that describes a disk or diskette volume.

direct access. (1) The access method used to READ or WRITE records on a disk or diskette device by specifying their location relative the beginning of the data set or volume. (2) In the Indexed Access Method, locating any record via its key without respect to the previous operation. (3) A condition in terminal I/O where a READTEXT or a PRINTTEXT is directed to a buffer which was previously enqueued upon by an IOCB.

directory. (1) A series of contiguous records in a volume that describe the contents in terms of allocated data sets and free space. (2) A series of contiguous records on a device that describe the contents in terms of allocated volumes and free space. (3) For the Indexed Access Method Version 2, a data set that defines the relationship between primary and secondary indexed files (secondary index support).

directory control entry (DCE). The first 32 bytes of the first record of a directory in which a description of the directory is stored.

directory member entry (DME). A 32-byte directory entry describing an allocated data set or volume.

display station. An IBM 4978, 4979, or 3101 display terminal or similar terminal with a keyboard and a video display.

DME. See directory member entry.

DSCB. See data set control block.

dynamic storage. An increment of storage that is appended to a program when it is loaded.

end-of-data indicator. A code that signals that the last record of a data set has been read or written. End-of-data is determined by an end-of-data pointer in the DME or by the physical end of the data set.

ECB. See event control block.

EDL. See Event Driven Language.

emulator. The portion of the Event Driven Executive supervisor that interprets EDL instructions and performs the function specified by each EDL statement.

end-of-tape (EOT). A reflective marker placed near the end of a tape and sensed during output. The marker signals that the tape is nearly full.

enter key. The key on the display terminal keyboard that, if pressed, tells the operating system to read the information you entered.

event control block (ECB). A control block used to record the status (occurred or not occurred) of an event; often used to synchronize the execution of tasks. ECBs are used in conjunction with the WAIT and POST instructions.

Event Driven Language (EDL). The language for input to the Event Driven Executive compiler (\$EDXASM), or the Macro and Host assemblers in conjunction with the Event Driven Executive macro libraries. The output is interpreted by the Event Driven Executive emulator.

EXIO (execute input or output). An EDL facility that provides user controlled access to Series/1 input/output devices.

external label. A label attached to the outside of a tape that identifies the tape visually. It usually contains items of identification such as file name and number, creation data, number of volumes, department number, and so on.

external name (EXTRN). The 1- to 8-character symbolic EBCDIC name for an entry point or data field that is not defined within the module that references the name.

FCA. See file control area.

FCB. See file control block.

file. A set of related records treated as a logical unit. Although file is often used interchangeably with data set, it usually refers to an indexed or a sequential data set.

file control area (FCA). A Multiple Terminal Manager data area that describes a file access request.

file control block (FCB). The first block of an indexed file. It contains descriptive information about the data contained in the file.

file control block extension. The second block of an indexed file. It contains the file definition parameters used to define the file.

file manager. A collection of subroutines contained within the program manager of the Multiple Terminal Manager that provides common support for all disk data transfer operations as needed for transaction-oriented application programs. It supports indexed and direct files under the control of a single callable function.

floating point. A positive or negative number that can have a decimal point.

formatted screen image. A collection of display elements or display groups (such as operator prompts and field input names and areas) that are presented together at one time on a display device.

free pool. In an indexed data set, a group of blocks that can be used for either data blocks or index blocks. These differ from other free blocks in that these are not initially assigned to specific logical positions in the file.

free space. In an indexed file, records blocks that do not currently contain data, and are available for use.

free space entry (FSE). An 8-byte directory entry defining an area of free space within a volume or a device.

FSE. See free space entry.

general purpose interface bus. The IEEE Standard 488-1975 that allows various interconnected devices to be attached to the GPIB adapter (RPQ D02118).

Glossary of Terms and Abbreviations

GPIB. See general purpose interface bus.

group. A unit of 100 records in the spool data set allocated to a spool job.

H exchange format. A standard format for exchanging data on diskettes between systems or devices.

host assembler. The assembler licensed program that executes in a 370 (host) system and produces object output for the Series/1. The source input to the host assembler is coded in Event Driven Language or Series/1 assembler language. The host assembler refers to the System/370 Program Preparation Facility (5798-NNQ).

host system. Any system whose resources are used to perform services such as program preparation for a Series/1. It can be connected to a Series/1 by a communications link.

IACB. See indexed access control block.

IAR. See instruction address register.

ICB. See indexed access control block.

IIB. See interrupt information byte.

image store. The area in a 4978 that contains the character image table.

immediate data. A self-defining term used as the operand of an instruction. It consists of numbers, messages or values which are processed directly by the computer and which do not serve as addresses or pointers to other data in storage.

index. In an indexed file, an ordered collection of pairs of keys and pointers, used to sequence and locate records.

index block. In an indexed file, an area that contains control information and index entries. These blocks are a multiple of 256 bytes.

indexed access control block (IACB/ICB). The control block that relates an application program to an indexed file.

indexed access method. An access method for direct or sequential processing of fixed-length records by use of a record's key.

indexed data set. Synonym for indexed file.

indexed file. A file specifically created, formatted and used by the Indexed Access Method. An indexed file is sometimes called an indexed data set.

index entry. In an indexed file, a key-pointer pair, where the pointer is used to locate a lower-level index block or a data block.

index register (#1, #2). Two words defined in EDL and contained in the task control block for each task. They are used to contain data or for address computation.

input buffer. (1) See buffer. (2) In the Multiple Terminal Manager, an area for terminal input and output.

input output control block (IOCB). A control block containing information about a terminal such as the symbolic name, size and shape of screen, the size of the forms in a printer, or an optional reference to a user provided buffer.

instruction address register (IAR). The pointer that identifies the machine instruction currently being executed. The Series/1 maintains a hardware IAR to determine the Series/1 assembler instruction being executed. It is located in the level status block (LSB).

integer. A positive or negative number that has no decimal point.

interactive. The mode in which a program conducts a continuous dialogue between the user and the system.

internal label. An area on tape used to record identifying information (similar to the identifying information placed on an external label). Internal labels are checked by the system to ensure that the correct volume is mounted.

interrupt information byte (IIB). In the Multiple Terminal Manager, a word containing the status of a previous input/output request to or from a terminal.

invoke. To load and activate a program, utility, procedure, or subroutine into storage so it can run.

job. A collection of related program execution requests presented in the form of job control statements, identified to the jobstream processor by a JOB statement.

job control statement. A statement in a job that specifies requests for program execution, program parameters, data set definitions, sequence of execution, and, in general, describes the environment required to execute the program.

job stream processor. The job processing facility that reads job control statements and processes the requests made by these statements. The Event Driven Executive job stream processor is \$JOBUTIL.

jumper. (1) A wire or pair of wires which are used for the arbitrary connection between two circuits or pins in an attachment card. (2) To connect wire(s) to an attachment card or to connect two circuits.

key. In the Indexed Access Method, one or more consecutive characters used to identify a record and establish its order with respect to other records. See also key field.

key field. A field, located in the same position in each record of an indexed file, whose content is used for the key of a record.

level status block (LSB). A Series/1 hardware data area that contains processor status. This area is eleven words in length.

library. A set of contiguous records within a volume. It contains a directory, data sets and/or available space.

line. A string of characters accepted by the system as a single input from a terminal; for example, all characters entered before the carriage return on the teletypewriter or the ENTER key on the display station is pressed.

link edit. The process of resolving external symbols in one or more object modules. A link edit is performed with \$EDXLINK whose output is a loadable program.

listener. A controller or active device on a GPIB bus that is configured to accept information from the bus.

load mode. In the Indexed Access Method, the mode in which records are loaded into base record slots in an indexed file.

load module. A single module having cross references resolved and prepared for loading into storage for execution. The module is the output of the \$UPDATE or \$UPDATEH utility.

load point. (1) Address in the partition where a program is loaded. (2) A reflective marker placed near the beginning of a tape to indicate where the first record is written.

lock. In the Indexed Access Method, a method of indicating that a record or block is in use and is not available for another request.

logical screen. A screen defined by margin settings, such as the TOPM, BOTM, LEFTM and RIGHTM parameters of the TERMINAL or IOCB statement.

LSB. See level status block.

mapped storage. The processor storage that you defined on the SYSTEM statement during system generation.

member. A term used to identify a named portion of a partitioned data set (PDS). Sometimes member is also used as a synonym for a data set. See data set.

menu. A formatted screen image containing a list of options. The user selects an option to invoke a program.

menu-driven. The mode of processing in which input consists of the responses to prompting from an option menu.

message. In data communications, the data sent from one station to another in a single transmission. Stations communication with a series of exchanged messages.

multifile volume. A unit of recording media, such as tape reel or disk pack, that contains more than one data file.

multiple terminal manager. An Event Driven Executive licensed program that provides support for transaction-oriented applications on a Series/1. It provides the capability to define transactions and manage the programs that support those transactions. It also manages multiple terminals as needed to support these transactions.

multivolume file. A data file that, due to its size, requires more than one unit of recording media (such as tape reel or disk pack) to contain the entire file.

new high key. A key higher than any other key in an indexed file.

nonlabeled tapes. Tapes that do not contain identifying labels (as in standard labeled tapes) and contain only files separated by tapemarks.

null character. A user-defined character used to define the unprotected fields of a formatted screen.

option selection menu. A full screen display used by the Session Manager to point to other menus or system functions, one of which is to be selected by the operator. (See primary option menu and secondary option menu.)

output buffer. (1) See buffer. (2) In the Multiple Terminal Manager, an area used for screen output and to pass data to subsequent transaction programs.

overlay. The technique of reusing a single storage area allocated to a program during execution. The storage area can be reused by loading it with overlay programs that have been specified in the PROGRAM statement of the program or by calling overlay segments that have been specified in the OVERLAY statement of \$EDXLINK.

overlay area. A storage area within a program reserved for overlay programs specified in the PROGRAM statement or overlay segments specified in the OVERLAY statement in \$EDXLINK.

overlay program. A program in which certain control sections can use the same storage location at different times during execution. An overlay program can execute concurrently as an asynchronous task with other programs and is specified in the EDL PROGRAM statement in the main program.

overlay segment. A self-contained portion of a program that is called and sequentially executes as a synchronous task. The entire program that calls the overlay segment need not be maintained in storage while the overlay segment is executing. An overlay segment is specified in the OVERLAY statement of \$EDXLINK or \$XPSLINK (for initialization modules).

overlay segment area. A storage area within a program or supervisor reserved for overlay segments. An overlay segment area is specified with the OVLAREA statement of \$EDXLINK.

Glossary of Terms and Abbreviations

parameter selection menu. A full screen display used by the Session Manager to indicate the parameters to be passed to a program.

partition. A contiguous fixed-sized area of storage. Each partition is a separate address space.

performance volume. A volume whose name is specified on the DISK definition statement so that its address is found during IPL, increasing system performance when a program accesses the volume.

physical timer. Synonym for timer (hardware).

polling. In data communications, the process by which a multipoint control station asks a tributary if it can receive messages.

precision. The number of words in storage needed to contain a value in an operation.

prefind. To locate the data sets or overlay programs to be used by a program and to store the necessary information so that the time required to load the prefound items is reduced.

primary file. An indexed file containing the data records and primary index.

primary file entry. For the Indexed Access Method Version 2, an entry in the directory describing a primary file.

primary index. The index portion of a primary file. This is used to access data records when the primary key is specified.

primary key. In an indexed file, the key used to uniquely identify a data record.

primary-level index block. In an indexed file, the lowest level index block. It contains the relative block numbers (RBNs) and high keys of several data blocks. See cluster.

primary menu. The program selection screen displayed by the Multiple Terminal Manager.

primary option menu. The first full screen display provided by the Session Manager.

primary station. In a Series/1 to Series/1 attachment, the processor that control communication between the two computers. Contrast with secondary station.

primary task. The first task executed by the supervisor when a program is loaded into storage. It is identified by the PROGRAM statement.

priority. A combination of hardware interrupt level priority and a software ranking within a level. Both primary and secondary tasks will execute asynchronously within the system according to the priority assigned to them.

process mode. In the Indexed Access Method, the mode in which records can be retrieved, updated, inserted or deleted.

processor status word (PSW). A 16-bit register used to (1) record error or exception conditions that may prevent further processing and (2) hold certain flags that aid in error recovery.

program. A disk- or diskette-resident collection of one or more tasks defined by a PROGRAM statement; the unit that is loaded into storage. (See primary task and secondary task.)

program header. The control block found at the beginning of a program that identifies the primary task, data sets, storage requirements and other resources required by a program.

program/storage manager. A component of the Multiple Terminal Manager that controls the execution and flow of application programs within a single program area and contains the support needed to allow multiple operations and sharing of the program area.

protected field. A field in which the operator cannot use the keyboard to enter, modify, or erase data.

PSW. See processor status word.

QCB. See queue control block.

QD. See queue descriptor.

QE. See queue element.

queue control block (QCB). A data area used to serialize access to resources that cannot be shared. See serially reusable resource.

queue descriptor (QD). A control block describing a queue built by the DEFINEQ instruction.

queue element (QE). An entry in the queue defined by the queue descriptor.

quiesce. To bring a device or a system to a halt by rejection of new requests for work.

quiesce protocol. A method of communication in one direction at a time. When sending node wants to receive, it releases the other node from its quiesced state.

record. (1) The smallest unit of direct access storage that can be accessed by an application program on a disk or diskette using READ and WRITE. Records are 256 bytes in length. (2) In the Indexed Access Method, the logical unit that is transferred between \$IAM and the user's buffer. The length of the buffer is defined by the user. (3) In BSCAM communications, the portions of data transmitted in a message. Record length (and, therefore, message length) can be variable.

recovery. The use of backup data to recreate data that has been lost or damaged.

reflective marker. A small adhesive marker attached to the reverse (nonrecording) surface of a reel of magnetic tape. Normally, two reflective markers are used on each reel of tape. One indicates the beginning of the recording area on the tape (load point), and the other indicates the proximity to the end of the recording area (EOT) on the reel.

relative block address (RBA). The location of a block of data on a 4967 disk relative to the start of the device.

relative record number. An integer value identifying the position of a record in a data set relative to the beginning of the data set. The first record of a data set is record one, the second is record two, the third is record three.

relocation dictionary (RLD). The part of an object module or load module that is used to identify address and name constants that must be adjusted by the relocating loader.

remote management utility control block (RCB). A control block that provides information for the execution of remote management utility functions.

reorganize. The process of copying the data in an indexed file to another indexed file in a manner that rearranges the data for more optimum processing and free space distribution.

restart. Starting the spool facility w the spool data set contains jobs from a previous session. The jobs in the spool data set can be either deleted or printed when the spool facility is restarted.

return code. An indicator that reflects the results of the execution of an instruction or subroutine. The return code is usually placed in the task code word (at the beginning of the task control block).

roll screen. A display screen which is logically segmented into an optional history area and a work area. Output directed to the screen starts display at the beginning of the work area and continues on down in a line-by-line sequence. When the work area gets full, the operator presses ENTER/SEND and its contents are shifted into the optional history area and the work area itself is erased. Output now starts again at the beginning of the work area.

SBIOCB. See sensor based I/O control block.

second-level index block. In an indexed data set, the second-lowest level index block. It contains the addresses and high keys of several primary-level index blocks.

secondary file. See secondary index.

secondary index. For the Indexed Access Method Version 2, an indexed file used to access data records by their secondary keys. Sometimes called a secondary file.

secondary index entry. For the Indexed Access Method Version 2, this an an entry in the directory describing a secondary index.

secondary key. For the Indexed Access Method Version 2, the key used to uniquely identify a data record.

secondary option menu. In the Session Manager, the second in a series of predefined procedures grouped together in a hierarchical structure of menus. Secondary option menus provide a breakdown of the functions available under the session manager as specified on the primary option menu.

secondary task. Any task other than the primary task. A secondary task must be attached by a primary task or another secondary task.

secondary station. In a Series/1 to Series/1 attachment, the processor that is under the control of the primary station.

sector. The smallest addressable unit of storage on a disk or diskette. A sector on a 4962 or 4963 disk is equivalent to an Event Driven Executive record. On a 4964 or 4966 diskette, two sectors are equivalent to an Event Driven Executive record.

selection. In data communications, the process by which the multipoint control station asks a tributary station if it is ready to send messages.

self-defining term. A decimal, integer, or character that the computer treats as a decimal, integer, or character and not as an address or pointer to data in storage.

sensor based I/O control block (SBIOCB). A control block containing information related to sensor I/O operations.

sequential access. The processing of a data set in order of occurrence of the records in the data set. (1) In the Indexed Access Method, the processing of records in ascending collating sequence order of the keys. (2) When using READ/WRITE, the processing of records in ascending relative record number sequence.

serially reusable resource (SRR). A resource that can only be accessed by one task at a time. Serially reusable resources are usually managed via (1) a QCB and ENQ/DEQ statements or (2) an ECB and WAIT/POST statements.

service request. A device generated signal used to inform the GPIB controller that service is required by the issuing device.

session manager. A series of predefined procedures grouped together as a hierarchical structure of menus from which you select the utility functions, program preparation facilities, and language processors needed to prepare and execute application programs. The menus consist of a primary option menu that displays functional groupings and secondary option menus that display a breakdown of these functional groupings.

shared resource. A resource that can be used by more than one task at the same time.

Glossary of Terms and Abbreviations

shut down. See data set shut down.

source module/program. A collection of instructions and statements that constitute the input to a compiler or assembler. Statements may be created or modified using one of the text editing facilities.

spool job. The set of print records generated by a program (including any overlays) while engaged to a printer designated as a spool device.

spool session. An invocation and termination of the spool facility.

spooling. The reading of input data streams and the writing of output data streams on storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

SRQ. See service request.

stand-alone dump. An image of processor storage written to a diskette.

stand-alone dump diskette. A diskette supplied by IBM or created by the \$DASDI utility.

standard labels. Fixed length 80-character records on tape containing specific fields of information (a volume label identifying the tape volume, a header label preceding the data records, and a trailer label following the data records).

static screen. A display screen formatted with predetermined protected and unprotected areas. Areas defined as operator prompts or input field names are protected to prevent accidental overlay by input data. Areas defined as input areas are not protected and are usually filled in by an operator. The entire screen is treated as a page of information.

station. In BSCAM communications, a BSC line attached to the Series/1 and functioning in a point-to-point or multipoint connection. Also, any other terminal or processor with which the Series/1 communicates.

subroutine. A sequence of instructions that may be accessed from one or more points in a program.

supervisor. The component of the Event Driven Executive capable of controlling execution of both system and application programs.

system configuration. The process of defining devices and features attached to the Series/1.

SYSGEN. See system generation.

system generation. The processing of defining I/O devices and selecting software options to create a supervisor tailored to the needs of a specific Series/1 hardware configuration and application.

system partition. The partition that contains the root segment of the supervisor (partition number 1, address space 0).

talker. A controller or active device on a GPIB bus that is configured to be the source of information (the sender) on the bus.

tape device data block (TDB). A resident supervisor control block which describes a tape volume.

tapemark. A control character recorded on tape used to separate files.

task. The basic executable unit of work for the supervisor. Each task is assigned its own priority and processor time is allocated according to this priority. Tasks run independently of each other and compete for the system resources. The first task of a program is the primary task. All tasks attached by the primary task are secondary tasks.

task code word. The first two words (32 bits) of a task's TCB; used by the emulator to pass information from system to task regarding the outcome of various operations, such as event completion or arithmetic operations.

task control block (TCB). A control block that contains information for a task. The information consists of pointers, save areas, work areas, and indicators required by the supervisor for controlling execution of a task.

task supervisor. The portion of the Event Driven Executive that manages the dispatching and switching of tasks.

TCB. See task control block.

terminal. A physical device defined to the EDX system using the TERMINAL configuration statement. EDX terminals include directly attached IBM displays, printers and devices that communicate with the Series/1 in an asynchronous manner.

terminal control block (CCB). A control block that defines the device characteristics, provides temporary storage, and contains links to other system control blocks for a particular terminal.

terminal environment block (TEB). A control block that contains information on a terminal's attributes and the program manager operating under the Multiple Terminal Manager. It is used for processing requests between the terminal servers and the program manager.

terminal screen manager. The component of the Multiple Terminal Manager that controls the presentation of screens and communications between terminals and transaction programs.

terminal server. A group of programs that perform all the input/output and interrupt handling functions for terminal devices under control of the Multiple Terminal Manager.

terminal support. The support provided by EDX to manage and control terminals. See terminal.

timer. The timer features available with the Series/1 processors. Specifically, the 7840 Timer Feature card (4955 only) or the native timer (4952, 4954, and 4956). Only one or the other is supported by the Event Driven Executive.

trace range. A specified number of instruction addresses within which the flow of execution can be traced.

transaction oriented applications. Program execution driven by operator actions, such as responses to prompts from the system. Specifically, applications executed under control of the Multiple Terminal Manager.

transaction program. See transaction-oriented applications.

transaction selection menu. A Multiple Terminal Manager display screen (menu) offering the user a choice of functions, such as reading from a data file, displaying data on a terminal, or waiting for a response. Based upon the choice of option, the application program performs the requested processing operation.

tributary station. In BSCAM communications, the stations under the supervision of a control station in a multipoint connection. They respond to the control station's polling and selection.

unmapped storage. The processor storage in your processor that you did not define on the SYSTEM statement during system generation.

unprotected field. A field in which the operator can use the keyboard to enter, modify or erase data. Also called non-protected field.

update. (1) To alter the contents of storage or a data set. (2) To convert object modules, produced as the output of an assembly or compilation, or the output of the linkage editor, into a form that can be loaded into storage for program execution and to update the directory of the volume on which the loadable program is stored.

user exit. (1) Assembly language instructions included as part of an EDL program and invoked via the USER instruction. (2) A point in an IBM-supplied program where a user written routine can be given control.

variable. An area in storage, referred to by a label, that can contain any value during program execution.

vary offline. (1) To change the status of a device from online to offline. When a device is offline, no data set can be accessed on that device. (2) To place a disk or diskette in a state where it is unknown by the system.

vary online. To place a device in a state where it is available for use by the system.

vector. An ordered set or string of numbers.

volume. A disk, diskette, or tape subdivision defined using \$INITDSK or \$TAPEUT1.

volume descriptor entry (VDE). A resident supervisor control block that describes a volume on a disk or diskette.

volume label. A label that uniquely identifies a single unit of storage media.

Index

The following index contains entries for this book only. See the *Library Guide and Common Index* for a Common Index to all Event Driven Executive books.

Special Characters

\$\$EDXIT, interpreting the output MC-384
\$\$EDXIT, sample output MC-384
\$DISKUT3 program
 return codes MC-310
\$EDXASM Event Driven Language compiler
 completion codes MC-290
\$EDXLINK linkage editor
 completion codes MC-291
\$IAM
 return codes MC-315
\$IMDATA subroutine
 return codes MC-313
\$IMOPEN subroutine
 return codes MC-313
\$IMPROT subroutine
 return codes MC-313
\$JOBUTIL job stream processor
 completion codes MC-292
\$PDS
 return codes MC-323
\$UPDATE utility
 completion codes MC-293

A

ACCA
 return codes MC-345
address, failing instruction MC-380
auto IPL, description MC-383

B

binary synchronous communications (BSC)
 return codes MC-303
bit descriptions, PSW MC-382
 auto IPL indicator (bit 13) MC-383
 floating-point exception (bit 5) MC-383
 I/O check (bit 11) MC-383
 invalid function (bit 4) MC-383
 invalid storage address (bit 1) MC-382
 power/thermal warning indicator (bit 15) MC-384
 privilege violate (bit 2) MC-383
 processor control check (bit 10) MC-383
 protect check (bit 3) MC-383
 sequence indicator check (bit 12) MC-383
 specification check (bit 0) MC-382
 stack exception (bit 6) MC-383
 storage parity (bit 8) MC-383
 translator enabled indicator (bit 14) MC-383
bit settings
 processor status word MC-382
boundary
 violations MC-382

Index

C

channel attach
 post codes MC-297
 return codes MC-305
class interrupt descriptions MC-382
codes
 completion MC-288
 overview MC-287
 post MC-294
 return MC-300
completion codes
 \$EDXASM MC-290
 \$EDXLINK MC-291
 \$JOBUTIL MC-292
 \$UPDATE MC-293
 cross partition supervisor MC-294
 overview MC-288
cross-partition supervisor
 completion codes MC-294

D

data formatting
 return codes MC-307
disk
 read/write return codes MC-308
diskette
 read/write return codes MC-308

E

EXIO
 return codes MC-311

F

floating point
 return codes MC-312
floating-point
 exception, description MC-383

G

general purpose interface bus
 return codes MC-314

H

Host Communications Facility
 return codes MC-347

I

I/O check, description MC-383
instruction address, failing MC-380
interprocessor communications
 return codes MC-345
interrupt descriptions, class MC-382
invalid function, description MC-383
invalid storage address, description MC-382

J

job queue processor
 return codes MC-318

L

LOAD
 return codes MC-319

M

message handler
 return codes MC-320
Multiple Terminal Manager
 return codes MC-321

N

NETBIND
 return codes MC-337
NETCLOSE
 return codes MC-337
NETCTL instruction
 return codes MC-330
NETGET instruction
 return codes MC-332
NETINIT instruction
 return codes MC-334
NETOPEN
 return codes MC-338
NETPUT instruction
 return codes MC-335
NETRECV
 return codes MC-338
NETSEND
 return codes MC-339
NETTERM instruction
 return codes MC-336
NETUBND
 return codes MC-341

P

post codes

- channel attach MC-297
- overview MC-294
- SNA BIND event MC-296
- tape MC-299
- WAITM instruction MC-300

power/thermal warning, description MC-384

privilege violate, description MC-383

processor control check, description MC-383

processor status word

- converting to bits MC-381
- how to interpret MC-381

program check

- \$\$EDXIT program check MC-384
- application program check MC-379
- bit settings, interpreting PSW MC-382
- locating failing instruction MC-380
- message description MC-380
- message types MC-379
 - \$\$EDXIT error exit MC-384
 - application check MC-379
 - system check MC-380
- message, examples MC-379
- overview MC-379
- processor status word, analysis MC-381
- register contents at failure MC-380
- system program check MC-379

protect check, description MC-383

PSW bits, interpreting MC-382

PSW, interpreting MC-381

R

register contents, program check MC-380

return codes

- \$DISKUT3 MC-310
- \$IAM MC-315
- \$IMDATA subroutine MC-313
- \$IMOPEN subroutine MC-313
- \$IMPROT MC-313
- \$JOBQUT MC-318
- \$PDS MC-323
- \$\$SUBMIT MC-318
- \$\$SUBMITP MC-318
- ACCA MC-345
- binary synchronous communications MC-303
- channel attach MC-305
- data formatting MC-307
- disk and diskette read/write MC-308
- EXIO MC-311
- floating point MC-312
- GPIB MC-314
- Host Communications Facility MC-347
- interprocessor communications MC-345
- job queue processor MC-318
- LOAD MC-319
- message handler MC-320

MTM MC-321

NETBIND MC-337

NETCLOSE MC-337

NETCTL MC-330

NETGET MC-332

NETINIT MC-334

NETOPEN MC-338

NETPUT MC-335

NETRECV MC-338

NETSEND MC-339

NETTERM MC-336

NETUBND MC-341

overview MC-300

SDLC MC-325

sensor-based I/O MC-324

Series/1-to-Series/1 attachment MC-326

session termination (SNA) MC-342

sort/merge MC-327

spool MC-328

system MC-329

tape read/write MC-343

terminal I/O MC-344

unmapped storage MC-349

X.21 Circuit Switched Network MC-350

4975 printer MC-346

S

SDLC communications

- return codes MC-325

sensor-based I/O

- return codes MC-324

sequence indicator error, description MC-383

Series/1-to-Series/1 attachment

- return codes MC-326

SNA BIND event

- post codes MC-296

sort/merge

- return codes MC-327

specification check, description MC-382

spooling

- return codes MC-328

stack exception, description MC-383

standard program check message, formats MC-379

stop codes

- meanings MC-376

- obtaining MC-375

storage parity error, description MC-383

T

tape

- post codes MC-299

- read/write return codes MC-343

task error exit routine

- \$\$EDXIT MC-384

terminal I/O

- return codes MC-344

Index

translator enabled, description MC-383

U

unmapped storage
return codes MC-349

V

virtual terminal
return codes MC-345

W

WAITM instruction
post codes MC-300

X

X.21 circuit switched network
return codes MC-350

4

4975 printer
return codes MC-346

**IBM Series/1 Event Driven Executive
Messages and Codes**

Order No. SC34-0445-1

**READER'S
COMMENT
FORM**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

Please Do Not Staple

Fold and tape

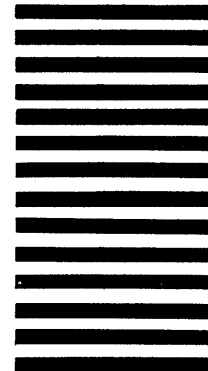


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Information Development, Department 27T
P.O. Box 1328
Boca Raton, Florida 33432



Fold and tape

Please Do Not Staple

Fold and tape





International Business Machines Corporation

SC34-0445-1

Program Numbers: 5719-XS4, 5719-XX5, 5719-ASA,
5719-LMS, 5719-LM5, 5719-PGH, 5719-AM4, 5719-CX1,
5719-MS2, 5719-SM2, 5719-SX1, and 5719-SX2

File No.: S1-40

Printed in U.S.A.