

NOS/VE

Diagnostic Messages

Usage

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Manual History

Revision	System Version	PSR Level	Date
A	1.02	-	February 1984
B	1.1.1	613	July 1984
C	1.1.3	644	January 1986
D	1.1.4	649	Online manual only
E	1.2.1	664	October 1986
F	1.2.1	670	January 1987
G	1.2.2	678	July 1987
H	1.2.3	688	September 1987
J	1.3.1	700	April 1988
K	1.4.1	716	December 1988

Revision K of this manual, printed December 1988, reflects NOS/VE Version 1.4.1 at PSR level 716.

The organization of this manual is in alphabetical and numerical order by the condition code, which consists of the two-character product identifier followed by one to four digits. The appropriate diagnostic message listing accompanies each condition code entry.

Due to extensive changes, change bars are not used and all pages reflect the latest revision level. This manual obsoletes all previous editions.

©1984, 1986, 1987, 1988 by Control Data Corporation
All rights reserved.
Printed in the United States of America.

Contents

About This Manual	5	Diagnostic Description Format	1-1
Audience	5	Message Severity Levels	1-2
Organization	5	Product Identifiers	1-3
Conventions	5	Diagnostic Message Listing	2-1
Submitting Comments	6	Related Manuals	A-1
CYBER Software Support		Ordering Printed Manuals	A-1
Hotline	6	Accessing Online Manuals	A-1
How to Use the Diagnostic			
Listing	1-1		
Introduction	1-1		

Tables

A-1. Related Manuals	A-2
----------------------------	-----

About This Manual

This manual provides a sorted listing of diagnostic messages issued by the CONTROL DATA® Network Operating System/Virtual Environment (NOS/VE) and its product set. It does not include messages produced by the ANALYZE_CDCNET_DUMP Utility. Those messages are documented in the CDCNET Network Analysis manual and the CDCNET online messages manual.

Audience

This manual is written for the applications programmer. It provides you with descriptions and user actions for each message. In some cases you are directed to another manual for further information. It is assumed you are familiar with the System Command Language and error processing in general.

Organization

Chapter 1 provides general information about NOS/VE diagnostic messages and how to use the listing in chapter 2. Chapter 2 lists the diagnostics in alphabetical and numerical order by condition code. Appendix A lists NOS/VE manuals and related hardware manuals, and provides information on ordering printed copies and accessing online versions.

Conventions

The following conventions are used in this manual:

- | | |
|-----------|--|
| UPPERCASE | Uppercase is used for names of commands, functions, and parameters (and their abbreviations). Uppercase is also used for names of variables, files, system constants, and terminal keys and function keys when they occur in text. |
| numbers | All numbers are decimal unless otherwise noted. |

Submitting Comments

There is a comment sheet at the back of this manual. You can use it to give us your opinion of the manual's usability, to suggest specific improvements, and to report errors. Mail your comments to:

Control Data
Technical Publications ARH219
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

Please indicate whether you would like a response.

If you have access to SOLVER, the Control Data online facility for reporting problems, you can use it to submit comments about the manual. When entering your comments, use NV0 (zero) as the product identifier. Include the name and publication number of the manual.

If you have questions about the packaging and/or distribution of a printed manual, write to:

Control Data
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103-2495

or call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

CYBER Software Support Hotline

Control Data's CYBER Software Support maintains a hotline to assist you if you have trouble using our products. If you need help not provided in the documentation, or find the product does not perform as described, call us at one of the following numbers. A support analyst will work with you.

From the USA and Canada: (800) 345-9903

From other countries: (612) 851-4131

How to Use the Diagnostic Listing **1**

Introduction	1-1
Diagnostic Description Format	1-1
Message Severity Levels	1-2
Product Identifiers	1-3

Introduction

All NOS/VE commands use a status record variable to report the completion status of system requests. To ensure that a consistent and meaningful response is presented to the command level user, a system message generator uses the status record to generate a formatted diagnostic message.

The organization of this manual assumes that the default information level for error messages is FULL for batch jobs. Refer to the LEVEL parameter of the CHANGE_MESSAGE_LEVEL command in the NOS/VE Commands and Functions manual for more information on setting message levels.

This manual is organized in alphabetical and numerical order by the condition code, which consists of the two-character product identifier followed by one to four digits.

Diagnostic Description Format

The following information is provided for each message.

- The severity level of the message. Severity levels are INFORMATIVE, CATASTROPHIC, ERROR, FATAL, and WARNING.
- The condition code.
- The full mode message text with variable fields represented as follows:
 - {file} Represents a file name or file path contained in the actual message.
 - {text} Represents replacement text contained in the actual message.
- The condition identifier, followed by a repetition of the condition code.
- A description of the message.
- Suggested user action to correct the problem.
- Further information (when applicable).

Condition identifiers (names) should be used instead of condition codes to test for specific conditions in SCL jobs and procedures. For example, condition code AA 5 has the condition identifier AAE\$GET_KEYS_THIS_OPEN. Refer to condition processing in the NOS/VE System Usage manual for more information on structuring jobs and testing for abnormal conditions. Also, see the \$CONDITION_NAME function in the NOS/VE Commands and Functions manual for information on determining and using the condition identifier in an SCL job.

Message Severity Levels

The severity of each NOS/VE message is indicated as follows:

Severity Level	Description
--(informative)--	These messages report conditions encountered during command processing that do not cause incorrect or incomplete operation of a command.
--WARNING--	These messages report conditions encountered during command processing that may have caused incorrect or incomplete operation of a command or of subsequent commands.
--ERROR--	These messages report that the last command was not completed correctly. By default, a batch job is terminated; for an interactive session, additional input is requested from the user to direct continued job processing.
--FATAL--	These messages report that the last command or subcommand was not completed correctly. Subsequent processing is usually provided in order to discover additional problems.
--CATASTROPHIC--	These messages report that the last command or subcommand was not completed correctly. No further processing for the requested function is possible.

Product Identifiers

The first two characters of the condition code and the condition name are identical and constitute the product identifier. The product identifiers and the product names for the messages contained in this manual are listed in the following table, along with the printed and online manuals that describe the products. Appendix A lists all NOS/VE manuals and related hardware manuals, and provides information on ordering printed manuals and accessing online manuals.

Product Identifier	Product Name	Manual Title	Online Manual
AA	Advanced Access	NOS/VE Advanced File Management Usage	AFM
		FORTTRAN Version 1 for NOS/VE Language Definition Usage	FORTTRAN
		CYBIL for NOS/VE Keyed-File and Sort/Merge Interfaces Usage	
AM	Access Method	CYBIL for NOS/VE Sequential and Byte-Addressable Files Usage	
AP	APL	APL for NOS/VE Language Definition Usage	
		APL for NOS/VE File Utilities Usage	
AV	Accounting/Validation	NOS/VE Accounting Analysis System Usage	
		NOS/VE Accounting and Validation Utilities for Dual State Usage	
BC	BASIC	BASIC for NOS/VE Usage	BASIC
CB	COBOL	COBOL for NOS/VE Usage	COBOL and EXAMPLES

Product Identifiers

Product Identifier	Product Name	Manual Title	Online Manual
CF	CYBIL Formatter	CYBIL for NOS/VE Language Definition Usage	CYBIL
CL	Command Language	NOS/VE Commands and Functions Quick Reference	SCL
		NOS/VE System Usage	EXAMPLES
		CYBIL for NOS/VE System Interface Usage	
CM	Configuration Management	NOS/VE System Performance and Maintenance Volume 2: Maintenance Usage	
CY	CYBIL	CYBIL for NOS/VE Language Definition Usage	CYBIL
DB	Debug Facility	Debug for NOS/VE Usage	DEBUG
		CYBIL for NOS/VE Language Definition Usage	CYBIL
		FORTRAN Version 1 for NOS/VE Language Definition Usage	FORTRAN
		BASIC for NOS/VE Usage	BASIC
		Pascal for NOS/VE Usage	PASCAL
		COBOL for NOS/VE Usage	COBOL and EXAMPLES
	C/VE for NOS/VE Reference Manual Usage	C	

Product Identifier	Product Name	Manual Title	Online Manual
DE	Desktop Environment	Desktop/VE for Macintosh Usage	
DF	File Server	NOS/VE File Server for STORNET and ESM-II Usage	
DU	Dump Analyzer	NOS/VE System Performance and Maintenance Volume 2: Maintenance Usage	ANALYZE_ DUMP
ES	Full Screen Editor	NOS/VE File Editor Tutorial/Usage	EXAMPLES
FA	File Migration Aids	Migration from NOS to NOS/VE Tutorial/Usage Migration from NOS to NOS/VE Standalone Tutorial/Usage Migration from NOS/BE to NOS/VE Tutorial/Usage Migration from NOS/BE to NOS/VE Standalone Tutorial/Usage	
FD	Format Display	NOS/VE Screen Formatting Usage Screen Design Facility for NOS/VE Usage	EXAMPLES SDF
FL	FORTTRAN Library	FORTTRAN Version 1 for NOS/VE Language Definition Usage	FORTTRAN

Product Identifier	Product Name	Manual Title	Online Manual
FM	File Management Utility	Migration from NOS to NOS/VE Tutorial/Usage	
		Migration from NOS to NOS/VE Standalone Tutorial/Usage	
		Migration from NOS/BE to NOS/VE Tutorial/Usage	
		Migration from NOS/BE to NOS/VE Standalone Tutorial/Usage	
		Migration from IBM to NOS/VE Tutorial/Usage	
		Migration from VAX/VMS to NOS/VE Tutorial/Usage	
		NOS/VE Advanced File Management Usage	AFM
FS	File System	CYBIL for NOS/VE File Management Usage	
FV	FORTTRAN Version 2	FORTTRAN Version 2 for NOS/VE Language Definition	
HU	Online Manuals	CYBER Online Text for NOS/VE Usage	CONTEXT
IC	Interstate Communication	CYBIL for NOS/VE System Interface Usage	

Product Identifier	Product Name	Manual Title	Online Manual
IF	Interactive Facility	CYBIL for NOS/VE File Management Usage	
		NOS/VE System Usage	EXAMPLES
IP	TCP/IP FTP	TCP/IP Usage	
JM	Job Management	CYBIL for NOS/VE System Interface Usage	
KT	Kermit	NOS/VE System Usage	KERMIT
LI	LISP	LISP for NOS/VE Language Definition Usage Supplement	
LL	Loader	CYBIL for NOS/VE System Interface Usage	
		NOS/VE Object Code Management Usage	OCM
ML	Math Library	Math Library for NOS/VE Usage	
		FORTRAN Version 1 for NOS/VE Language Definition Usage	FORTRAN
MM	Memory Management		
MV	Mail/VE Version 1	Mail/VE (Version 1) Usage	MAIL_VE
MS	Maintenance Services	NOS/VE System Performance and Maintenance Volume 2: Maintenance Usage	

Product Identifier	Product Name	Manual Title	Online Manual
NA	NAM/VE	NOS/VE Network Management Usage	
NF	Network File Transfer	NOS/VE Network Management Usage	
OC	Object Code Management	NOS/VE Object Code Management Usage	OCM
OF	Operator Facility	NOS/VE Operations Usage	
OS	Operating System	NOS/VE Operations Usage	
PA	Pascal	Pascal for NOS/VE Usage	PASCAL
PE	Programming Environment	Programming Environment for NOS/VE Summary	ENVIRON- MENT
PF	Permanent File Management	CYBIL for NOS/VE File Management Usage	
		NOS/VE System Usage	EXAMPLES
PM	Program Management	CYBIL for NOS/VE System Interface Usage	
PU	Permanent File Utilities	CYBIL for NOS/VE System Interface Usage	
		NOS/VE System Usage	EXAMPLES
QK	IM/Quick	IM/Quick for NOS/VE Tutorial	QUICK
RA	Release Administration		
RF	Remote Host Facility Access Method	NOS/VE LCN Configuration and Network Management	

Product Identifier	Product Name	Manual Title	Online Manual
RH	Remote Host Facility	NOS/VE System Usage Remote Host Facility Usage	
RM	Resource Management	CYBIL for NOS/VE File Management Usage	
SC	Source Code Utility	NOS/VE Source Code Management Usage	SCU and EXAMPLES
SF	Statistics Facility	NOS/VE System Performance and Maintenance Volume 1: Performance Usage	
SM	Sort/Merge	CYBIL for NOS/VE Keyed-File and Sort/Merge Interfaces Usage FORTRAN Version 1 for NOS/VE Language Definition Usage NOS/VE Advanced File Management Usage	FORTRAN AFM
ST	Edit Catalog	EDIT_CATALOG for NOS/VE Summary Introduction to NOS/VE Tutorial	EDIT_ CATALOG SCL
TU	Terminal Utility	NOS/VE Terminal Definition Usage	
XM	Xmodem		

--INFORMATIVE AA 0-- File {file} : {text} DELETE_KEYS done since last open.

Condition Identifier: AAE\$DELETE_KEYS_THIS_OPEN Code: AA 0

Description: Informative message stating the number of record deletion operations performed on the file since the last open.

User Action: None.

--INFORMATIVE AA 5-- File {file} : {text} GET_KEYS done since last open.

Condition Identifier: AAE\$GET_KEYS_THIS_OPEN Code: AA 5

Description: Informative message stating the total number of successful random read requests performed on the file since the last open.

User Action: None.

--INFORMATIVE AA 10-- File {file} : {text} GET_NEXT_KEYS done since last open.

Condition Identifier: AAE\$GET_NEXT_KEYS_THIS_OPEN Code: AA 10

Description: Informative message stating the total number of successful sequential read requests performed on the file since the last open.

User Action: None.

--INFORMATIVE AA 15-- File {file} : {text} PUT_KEYS (and PUTREPs->put) since last open.

Condition Identifier: AAE\$PUT_KEYS_THIS_OPEN Code: AA 15

Description: Informative message giving the total number of successful put requests performed since the last open. This number includes the puts performed by put/replace requests.

User Action: None.

--INFORMATIVE AA 20-- File {file} : {text} PUTREPs done since last open.

Condition Identifier: AAE\$PUTREPS_THIS_OPEN Code: AA 20

Description: Informative message giving the total number of successful put/replace requests performed since the last open.

User Action: None.

--INFORMATIVE AA 25-- File {file} : {text} REPLACE_KEYS (and PUTREPs->replace) since last open.

Condition Identifier: AAE\$REPLACE_KEYS_THIS_OPEN Code: AA 25

Description: Informative message giving the total number of successful replace requests performed on the file since the last open. This number includes the replace requests performed by put/replace requests.

User Action: None.

--INFORMATIVE AA 1000-- File{text} {file}{text} : {text} caused number of index levels to increase to{text}.

Condition Identifier: AAE\$ADDING_LEVEL_OF_INDEX Code: AA 1000

Description: Informative message indicating that the operation increased the number of index levels in the file. The number of index levels indicates the number of index blocks the system must search to access a data record.

User Action: None required. When the number of index levels exceeds two, you should consider re-creating the file with one or two index levels to improve performance.

Further Information: To learn how to re-create an indexed-sequential file, see the NOS/VE Advanced File Management Usage manual. To read a description of the indexed-sequential file data and index block structure, see the NOS/VE Advanced File Management Usage manual, the FORTRAN Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File & Sort/Merge Interfaces Usage manual.

--INFORMATIVE AA 1005-- File{text} {file}{text} : {text} has reached a file boundary : {text}.

Condition Identifier: AA\$FILE_BOUNDARY_ENCOUNTERED Code: AA 1005

Description: Informative message indicating that the operation encountered a file boundary (BOI or EOI).

User Action: None.

--WARNING AA 1010-- File{text} {file}{text} : {text} encountered operating system failure-- {text}

Condition Identifier: AA\$SYSTEM_ROUTINE_FAILED Code: AA 1010

Description: A system routine failed; therefore, the specified operation was not performed. A description of the result of the failure is listed. Processing continues.

User Action: Follow your site procedures for reporting software errors.

--INFORMATIVE AA 1015-- File{text} {file}{text} : the last error {text}

Condition Identifier: AA\$LAST_ERROR_REPEATED Code: AA 1015

Description: Informative message indicating that the last error of severity ERROR or FATAL for the file repeated the specified number of times.

User Action: Refer to the previous error message.

--INFORMATIVE AA 1020-- File{text} {file}{text} : last preceding non-fatal error follows:

Condition Identifier: AA\$PREVIOUS_NON_FATAL_ERROR Code: AA 1020

Description: Informative message repeating the complete diagnostic for the last nonfatal error to occur on the file before the current fatal error.

User Action: None.

--WARNING AA 1025-- File{text} {file}{text} has reached the size limit imposed by FILE_LIMIT.

Condition Identifier: AA\$FILE_REACHED_FILE_LIMIT Code: AA 1025

Description: No write requests can be performed because the file has reached its current maximum size as set by the file_limit attribute value.

User Action: To write additional records to the file, increase the file_limit attribute value with the CHANGE_FILE_ATTRIBUTES command.

Further Information: See the description of the CHANGE_FILE_ATTRIBUTE command in the NOS/VE System Usage manual.

--INFORMATIVE AA 1030-- File{text} {file}{text} : IFETCH of EX, KA, PKA, WSA, or DCT returns the value as an address, which is only useful for languages with pointers.

Condition Identifier: AA\$FETCHED_INFO_NOT_NECESSARY Code: AA 1030

Description: Fetching the key address (KA), primary-key address (PKA), or working storage address (WSA) value from the FIT is not appropriate because these values are stored as pointers which are not useful values in a FORTRAN program.

User Action: None required, although you should determine how the program is using the fetched value.

Further Information: See the Keyed-File Interface section in the FORTRAN Usage manual for more information on fetching FIT values.

--INFORMATIVE AA 1035-- File{text} {file}{text} : using MKL on an integer key is not recommended since the least significant portion of the integer will be ignored.

Condition Identifier: AA\$MKL_ON_INTEGER_KEY_DETECTED Code: AA 1035

Description: Specifying a major key length when the key_type is integer is not recommended. Major keys are intended for use with character keys (collated or uncollated).

User Action: None required.

--INFORMATIVE AA 1040-- File{text} {file}{text} : {text} is evicting previous data in the file.

Condition Identifier: AA\$EVICT_DATA_FROM_FILE Code: AA 1040

Description: Informative message stating that the data currently in the file is discarded so that the new keyed-file structure can be created.

User Action: None.

--INFORMATIVE AA 1045-- File{text} {file}{text} : begin deleting alternate key definitions.

Condition Identifier: AA\$BEGIN_DELETE_KEYDEFS Code: AA 1045

Description: Informative message indicating that the apply operation has begun processing the alternate-key deletion requests.

User Action: None.

--INFORMATIVE AA 1050-- File{text} {file}{text} : alternate key {text} has been deleted.

Condition Identifier: AA\$KEYDEF_HAS_BEEN_DELETED Code: AA 1050

Description: Informative message indicating that the alternate key has been successfully deleted.

User Action: None.

--INFORMATIVE AA 1055-- File{text} {file}{text} : end deleting alternate key definitions.

Condition Identifier: AA\$END_DELETE_KEYDEFS Code: AA 1055

Description: Informative message indicating that all pending alternate-key deletions have been processed.

User Action: None.

--INFORMATIVE AA 1060-- File{text} {file}{text} : begin creating labels for alternate key definitions.

Condition Identifier: AA\$BEGIN_ALTKEY_LABELS Code: AA 1060

Description: Informative message indicating that the apply operation has begun creating the internal file labels for the new alternate keys.

User Action: None.

--INFORMATIVE AA 1065-- File{text} {file}{text} : finished creating labels for alternate key definitions.

Condition Identifier: AA\$END_ALTKEY_LABELS Code: AA 1065

Description: Informative message indicating that the system has completed writing the internal labels for the new alternate keys.

User Action: None.

--INFORMATIVE AA 1070-- File{text} {file}{text} : data pass will not be done because this is a new file.

Condition Identifier: AA\$NO_DATA_PASS_FOR_NEW_FILE Code: AA 1070

Description: Informative message noting that the system does not attempt to collect alternate-key values when the file does not contain data.

User Action: None.

--INFORMATIVE AA 1075-- File{text} {file}{text} : begin collecting the alternate key values from the file.

Condition Identifier: AA\$BEGIN_AK_VALUE_COLLECTION Code: AA 1075

Description: Informative message stating the apply operation has begun collecting alternate-key values to build the alternate index. Each record is read to collect the values for all alternate-key definitions being applied.

User Action: None required.

--INFORMATIVE AA 1080-- File{text} {file}{text} : collection of the alternate key values is complete.

Condition Identifier: AA\$END_AK_VALUE_COLLECTION Code: AA 1080

Description: Informative message indicating that all alternate-key values have been collected for the new alternate index.

User Action: None.

--INFORMATIVE AA 1085-- File{text} {file}{text} : begin sorting the alternate key values.

Condition Identifier: AA\$BEGIN_ALTKEY_SORT Code: AA 1085

Description: Informative message indicating that the apply operation has begun sorting the alternate-key values for the new alternate index.

User Action: None.

--INFORMATIVE AA 1090-- File{text} {file}{text} : sorting of the alternate key values completed.

Condition Identifier: AA\$END_ALTKEY_SORT Code: AA 1090

Description: Informative message indicating that the system has completed sorting the alternate-key values.

User Action: None.

--INFORMATIVE AA 1095-- File{text} {file}{text} : begin building alternate key indexes into the file.

Condition Identifier: AA\$BEGIN_BUILDING_INDEXES Code: AA 1095

Description: Informative message indicating that the apply operation has begun building the indexes for the new alternate keys.

User Action: None.

--INFORMATIVE AA 1100-- File{text} {file}{text} : the {text} index is being built.

Condition Identifier: AA\$INDEX_BEING_BUILT Code: AA 1100

Description: Informative message indicating that the alternate index is now being built in the file.

User Action: None.

--INFORMATIVE AA 1105-- File{text} {file}{text} : AMP\$APPLY_KEY_DEFINITIONS completed building the alternate indexes into the file.

Condition Identifier: AA\$END_BUILDING_INDEXES Code: AA 1105

Description: Informative message indicating that the apply operation has completed building the requested alternate indexes.

User Action: None.

--INFORMATIVE AA 1110-- File{text} {file}{text} : Because this RMKDEF call specifies a 0 for KL (parameter position 4), it is assumed that this call and the next RMKDEF call specify an alternate key with sparse_key_control. Parameters 3 & 4 on the second RMKDEF call specify the SPARSE_KEY_CONTROL_POSITION.

Condition Identifier: AA\$START_170_STYLE_SPARSE_KEY Code: AA 1110

Description: Informative message stating that the RMKDEF call is expected to be the first of two RMKDEF calls that define an alternate key that uses sparse-key control.

User Action: None.

Further Information: The RMKDEF call is described in the FORTRAN Language Definition Usage manual.

--INFORMATIVE AA 1115-- Creating new file {file}.

Condition Identifier: AA\$CREATING_NEW_FILE Code: AA 1115

Description: Informative message stating that the command is creating a new file.

User Action: None required. If the file creation is unexpected (an existing file was to have been used), check the input file specification.

--INFORMATIVE AA 1120-- File {file} : By default, \$MAIN_FILE will be used by {text}.

Condition Identifier: AA\$MAIN_FILE_WILL_BE_USED Code: AA 1120

Description: If no nested file name is specified, the nested file is copied to the default nested file (\$MAIN_FILE) of the output file.

User Action: None required. If COPY_KEYED_FILE is to copy a nested file other than \$MAIN_FILE, specify the nested file name as the second value in the OUTPUT parameter value set, that is, (file_reference, nested_file_name).

--INFORMATIVE AA 1125-- File {file} : {text} has created alternate_key_name {text}.

Condition Identifier: AA\$RMKDEF_CREATED_ALT_KEY_NAME Code: AA 1125

Description: Informative message stating that the program has created an alternate key. The alternate-key name contains the RKW, RKP, and key length values used to create the key.

User Action: None.

Further Information: The RMKDEF call is described in the FORTRAN Language Definition Usage manual.

--INFORMATIVE AA 1130-- File{text} {file}{text} : No record met the criteria to qualify for an entry in the {text} index.

Condition Identifier: AA\$INDEX_IS_EMPTY Code: AA 1130

Description: The search for alternate-key values failed to find any that fit the criteria of the alternate-key definition. However, the alternate-key definition is stored with the file, and if records with alternate-key values that fit the criteria are written later to the file, the alternate index will be built then.

User Action: None required. Check that the alternate-key definition is correct.

Further Information: See the CREATE_KEY_DEFINITION subcommand description and the discussion of sparse-key control in the SCL Advanced File Management Usage manual.

--WARNING AA 1135-- {text} requires {file} to be an existing file.

Condition Identifier: AA\$FILE_DOES_NOT_EXIST Code: AA 1135

Description: The operation requires that the file be an existing file, that is, a file that has been previously opened.

User Action: Check that the correct file was specified.

--WARNING AA 1140-- File {file} : {text} cannot process a file with file organization = {text}.

Condition Identifier: AA\$FILE_IS_NOT_A_KEYED_FILE Code: AA 1140

Description: The file must be a keyed file, that is, its file_organization attribute must be defined as a keyed-file organization (indexed_sequential, direct_access, or system_key).

User Action: Check that the correct file is specified.

--WARNING AA 1145-- Pause_breaks are not permitted - to end processing, enter a terminate_break.

Condition Identifier: AA\$PAUSE_NOT_ALLOWED Code: AA 1145

Description: Entry of a pause_break_character does not stop processing.

User Action: If desired, enter the terminate_break_character (usually control-t) to stop processing.

--WARNING AA 1150-- File {file} : {text} found the {text} in block {text} contains an improper value for the {text}.

Condition Identifier: AA\$ILLEGAL_AAM_TABLE_VALUE Code: AA 1150

Description: Internal error; an internal routine found an incorrect value.

User Action: Check the structural properties of the file using a DISPLAY_KEYED_FILE_PROPERTIES command. If appropriate, report the error.

--WARNING AA 1155-- DISPLAY_KEYED_FILE detected errors in the keyed-file structure. See \$ERRORS and the output file for details.

Condition Identifier: AA\$BAD_AAM_FILE_STRUCTURE Code: AA 1155

Description: The DISPLAY_KEYED_FILE command found errors in the keyed-file structure. The error messages were written to the \$ERRORS file (by default, the OUTPUT file).

User Action: Read the messages written to the \$ERRORS file.

--WARNING AA 1160-- File {file} : {text} could not create a block cross_reference sort record for block {text} because {text}.

Condition Identifier: AA\$BAD_SORT_INPUT_RECORD Code: AA 1160

Description: Internal error; a sort attempt failed.

User Action: Report the software error.

--WARNING AA 1165-- File {file} : {text} received a bad block cross_reference record from Sort/Merge. The record contained {text}.

Condition Identifier: AA\$BAD_SORT_RECORD_RETURNED Code: AA 1165

Description: Internal error; a sort received a record with an invalid sort key.

User Action: Report the software error.

--WARNING AA 1170-- File {file} : {text} received the abnormal completion status {text}, from Sort/Merge.

Condition Identifier: AA\$SORT_RETURNED_BAD_STATUS Code: AA 1170

Description: Internal error; a sort attempt returned abnormal status.

User Action: Report the software error.

--WARNING AA 1175-- The {text} feature is not currently implemented.

Condition Identifier: AA\$FEATURE_NOT_IMPLEMENTED Code: AA 1175

Description: The request could not be executed as specified because a feature used by the request is not available with the current level of NOS/VE software.

User Action: Correct the request so that it does not require the unimplemented feature.

--WARNING AA 1180-- File{text} {file}{text} : The APPLY_KEY_DEFINITIONS subcommand in CREATE_ALTERNATE_INDEXES {text}.

Condition Identifier: AA\$ERRORS_IN_APPLY Code: AA 1180

Description: The alternate-key definition application found one or more nonfatal errors, but fewer errors than the error limit specified for the operation. (0 specifies unlimited errors). The nonfatal errors could include duplicate alternate-key values or short records.

User Action: None required, although you should examine the nonfatal error messages. If a nonfatal error should have terminated the apply, delete the alternate key, redefine it, and apply it with an error limit of 1 so that the first nonfatal error that occurs terminates the apply.

Further Information: See the APPLY_KEY_DEFINITIONS Subcommand description in the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1185-- File {text} in {file} : {text} cannot apply alternate key {text} from the input file. Part of the alternate key was eliminated by specifying nonembedded keys for the output file.

Condition Identifier: AAESALTKEY_PART_NEGATIVE_POS Code: AA 1185

Description: Because the input file has an embedded primary key and the output file has a nonembedded primary key, the primary-key value for the output file was taken from the beginning of each input record. The value was not written to the output record. An alternate key field included part of the removed value. Therefore, that alternate key was not built in the output file because the alternate-key values are no longer complete in the record data.

User Action: None required, although you should determine if the unbuilt alternate key is required for use of the output file.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1190-- File {file} : {text} encountered an error while attempting a sequential write operation (a PUT_NEXT). Part of the input data has been written to the output file.

Condition Identifier: AAESPARTIAL_COPY_OCCURRED Code: AA 1190

Description: COPY_KEYED_FILE could not complete the copy because of an error found by a write request.

User Action: Read the other messages for the cause of the error.

Further Information: The COPY_KEYED_FILE command is described in the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1195-- File {text} in {file} : {text} inserts the input key value at the beginning of the record when copying a nested file with nonembedded_ keys to a nested file with embedded_ keys. This changes the primary_key values if key_position is not zero.

Condition Identifier: AAESCOPYING_NON_EK_TO_EK Code: AA 1195

Description: When records are copied from a file with a nonembedded primary key to a file with an embedded primary key, the key value is inserted at the beginning of the record when it is written to the output file. If the primary-key value of the output file is to be the same as it was on the output file, the KEY_POSITION attribute value of the output file must be 0.

User Action: None required.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--INFORMATIVE AA 1200-- File {text} in {file} : {text} deletes the input key value from the beginning of the record when copying a nested file with embedded_ keys to a nested file with nonembedded_ keys. This changes the primary_key values if key_position is not zero.

Condition Identifier: AAESCOPYING_EK_TO_NON_EK Code: AA 1200

Description: When records are copied from a file with an embedded primary key to a file with a nonembedded primary key, the key value is always taken from the beginning of the record, regardless of the KEY_POSITION value of the input file. The key value is removed from the beginning of the data record when it is written to the output file.

User Action: None required.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1205-- You are doing APPLY_KEY_DEFINITIONS on file {file}. This is probably a long operation, and has been interrupted. Typing RUIN FILE now will cause an immediate stop, but the file will be ruined. Or you can type GO AHEAD, which will allow the operation to complete -- but thereafter, AAM will be forbidden to do anything except CLOSE.

Condition Identifier: AA\$TERMINATE_BREAK_ON_APPLY Code: AA 1205

Description: The user has entered a terminate_break character while an alternate-key definition is begin applied to the file. If the termination is carried out, the integrity of the keyed-file structure will be lost.

User Action: Enter a carriage return or any other entry except RUIN_FILE (uppercase or lowercase) to allow the application to continue.

Further Information: See the Remarks section in the APPLY_KEY_DEFINITIONS description in the NOS/VE Advanced File Management Usage manual or in the AMP\$APPLY_KEY_DEFINITIONS description in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--WARNING AA 1206-- You are doing APPLY_KEY_DEFINITIONS on file {file}. This is probably a long operation, and the job time limit has been reached. However, you can choose to continue by typing GO AHEAD now, which will allow the operation to complete. But this same interruption will recur after every ten seconds of additional CP time, until the operation completes. Thereafter, AAM will be forbidden to do anything except CLOSE. Or you can type RUIN FILE, which will immediately set the file ruined, and allow the time limit to terminate the job.

Condition Identifier: AA\$TIME_LIMIT_ON_APPLY Code: AA 1206

Description: The job time limit has been reached while an alternate-key is being applied to the file. If the operation is terminated, the integrity of the keyed-file structure will be lost.

User Action: Enter GO AHEAD to allow the application to continue.

--WARNING AA 1207-- File {file} : {text} was backed out, and hopefully the file is not ruined. But it has to be closed now.

Condition Identifier: AA\$LAST_UPDATE_BACKED_OUT Code: AA 1207

Description: The operation in progress has been interrupted. Instead of closing the file in a corrupted state, the operation has been backed out so that the file has been left untouched. However, the file is still open and must be closed before any further processing can continue on the file.

User Action: Close the file.

--WARNING AA 1210-- File {file} : A request to write a record specified a working_storage_length {text}. The record has been written using the fixed_record_length. If not enough data was in the working_storage_area, random data may have been written at the end of the record; if excess data was in the working_storage_area, the data was truncated. (This warning will not be repeated for any following write requests that specify an incorrect working_storage_length).

Condition Identifier: AA\$FIXED_REC_DIFFERS_FROM_MRL Code: AA 1210

Description: The working_storage_length on a write request did not match the MAXIMUM_RECORD_LENGTH attribute of the file. This may mean that the data provided for a write request was either too short or too long for a record.

All records written to the file are required to be the same length. So, if the record data was too long, it was truncated; if it was too short, the data following the record data was written at the end of the record.

This warning is issued only once.

User Action: Determine whether the error was in the data length specified on the write request or in the actual data to be written.

Check the records written to the file to determine if the correct data was written.

If appropriate, redefine the file with a different MAXIMUM_RECORD_LENGTH attribute value or with a variable-length RECORD_TYPE.

--WARNING AA 1215-- File{text} {file}{text} : The block size for the file is smaller than the system page size, forced-writing is enabled (the FORCED_WRITE attribute is not false), and shared access is possible. Under these conditions, a forced write of one block would also write any partly-altered blocks in the page. (Because the file is shared and the block size is smaller than the page size, more than one instance of open could be altering the same page). To avoid this situation, re-create the file with a block size equal to or greater than the page size.

Condition Identifier: AA\$PAGE_SIZE_AND_FORCED_WRITE Code: AA 1215

Description:

User Action: Do one or more of the following:

- Attach the file for exclusive access (share_modes=none).
- Change the FORCED_WRITE attribute value to FALSE.
- Re-create the file with a block length at least as long as the system page length. This is the default.

Further Information: The FORCED_WRITE and block length values are described in the FORTRAN Usage, CYBIL Keyed-File and Sort/Merge Interfaces, and NOS/VE Advanced File Management manuals.

Keyed-file sharing is described in the FORTRAN Usage and CYBIL Keyed-File and Sort/Merge Interfaces manuals.

The ATTACH_FILE command is described in the NOS/VE System Usage manual.

--WARNING AA 1220-- File{text} {file}{text} : AMP\$BEGIN_QUIET_PERIOD determined that the quiet period requested for this file is already in effect.

Condition Identifier: AA\$QUIET_PERIOD_ALREADY_ON Code: AA 1220

Description: An AMP\$BEGIN_QUIET_PERIOD call has already been issued for this instance of open and a subsequent AMP\$END_QUIET_PERIOD call has not been issued.

User Action: None required. If possible, correct the program so that it does not issue consecutive AMP\$BEGIN_QUIET_PERIOD calls without an intervening AMP\$END_QUIET_PERIOD call.

Further Information: To read about the quiet period calls, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--WARNING AA 1225-- File{text} {file}{text} : There is no quiet_period established for this file, so AMP\$BEGIN_QUIET_PERIOD did not process the request to end the quiet period.

Condition Identifier: AA\$QUIET_PERIOD_NOT_PRESENT Code: AA 1225

Description: An AMP\$END_QUIET_PERIOD call was issued without a preceding AMP\$BEGIN_QUIET_PERIOD call for the instance of open.

User Action: None required. If possible, change the program so that it does not issue an AMP\$END_QUIET_PERIOD call without a preceding AMP\$BEGIN_QUIET_PERIOD call.

Further Information: To read about the quiet period calls, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--INFORMATIVE AA 1230-- File {file} : Because the input file open position is \$EOI, {text} will copy the file attributes, but not the data records.

Condition Identifier: AA\$COPYING_ATTRIBUTES_ONLY Code: AA 1230

Description: The open position specified for the input file is \$EOI, its end-of-information. Thus, the copy begins at the end of the input file so no data records are copied. However, assuming the output file is a new file, the attributes not defined for the new output file are copied from the input file attribute set.

User Action: None required if the indicated operation was intended. Otherwise, if the data was to be copied, change the input file open position to \$BOI, its beginning-of-information, and repeat the copy operation.

Further Information: For a description of the COPY_KEYED_FILE command, see the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1235-- Either no output file was specified or the same file was specified as both the input file and the output file. {text} will overwrite file {file}, changing it directly.

Condition Identifier: AAE\$NO_OUTPUT_FILE_SPECIFIED Code: AA 1235

Description: Because no other output file was specified, the input file will be changed directly. Any changes made cannot be undone.

User Action: Check that the correct files were specified. If not, enter QUIT to end the session and then re-enter the CHANGE_KEYED_FILE command.

Further Information: For more information on the CHANGE_KEYED_FILE Utility, see the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1240-- File {file} : Because the input and output files specified on the command are the same file cycle, and the file is empty, CHANGE_KEYED_FILE will continue processing as if CREATE_KEYED_FILE had been entered.

Condition Identifier: AAE\$CREATING_KEYED_FILE Code: AA 1240

Description: The input file specified on the CHANGE_KEYED_FILE command is empty and so cannot be changed. Thus, the file will be processed as it would be for a CREATE_KEYED_FILE session.

User Action: Check that the correct input file was specified. If not, enter QUIT to end the session and enter the correct CHANGE_KEYED_FILE command.

Further Information: For more information on the CHANGE_KEYED_FILE utility, see the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1245-- File {file} : The output file is a direct_access file but, a value was not provided for INITIAL_HOME_BLOCK_COUNT. Please enter ADD_RECORDS commands in response to the prompt "IHBC/". Enter QUIT after specifying ALL files to be copied. Create_Keyed_File will then compute the INITIAL_HOME_BLOCK_COUNT and copy the records into the new file.

Condition Identifier: AAE\$PROMPTING_FOR_IHBC Code: AA 1245

Description: This message appears if the file_organization attribute is direct_access but the initial_home_block_count attribute has not been specified.

User Action: Enter ADD_RECORDS commands, specifying all files containing input records for the new file. All input files must be specified so that the command can calculate the correct home_block_count for the file.

Enter QUIT after all ADD_RECORDS commands have been entered. The command then computes the home_block_count, creates the new nested file, and copies the input records to the file.

Further Information: For further information on the CREATE_KEYED_FILE or CREATE_NESTED_FILE commands or direct-access file structure in general, see the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1250-- INITIAL_HOME_BLOCK_COUNT is {text}, based on LOADING_FACTOR = {text}, AVERAGE_RECORD_LENGTH = {text}, RECORD_COUNT = {text} and BLOCK_LENGTH = {text}.

Condition Identifier: AAE\$IHBC_VALUE Code: AA 1250

Description: This informative message gives the computed initial_home_block_count value and the values used to compute that value.

User Action: None required.

Further Information: For further information on direct-access file structure, see the NOS/VE Advanced File Management manual.

--WARNING AA 1255-- {text} requires SORT=TRUE when computing the INITIAL_HOME_BLOCK_COUNT so the SORT parameter value is ignored.

Condition Identifier: AA\$SORT_REQUIRED_FOR_IHBC Code: AA 1255

Description: The SORT parameter was specified as FALSE, but to compute the initial_home_block_count value, CREATE_KEYED_FILE must sort the input records. Therefore, SORT=FALSE is ignored.

User Action: None required.

--INFORMATIVE AA 1260-- To continue processing the remaining input records, enter: {text}{file}{text}.

Condition Identifier: AA\$ENTER_ASIS_TO_CONTINUE Code: AA 1260

Description: An error occurred, interrupting command processing. However, if you enter the indicated command, processing can continue after the record in error.

User Action: None required. If you do not enter the indicated command, only the records before the record in error are processed.

--WARNING AA 1265-- File {file} could not be copied because it was empty or positioned at its end_of_information.

Condition Identifier: AA\$FILE_EMPTY_OR_AT_EOI Code: AA 1265

Description: To be copied, a file must contain data to be copied and the file cannot be positioned at its end.

User Action: Check that the correct file was specified. If appropriate, change its file position to \$BOI.

--WARNING AA 1270-- File {file} : {text} encountered an error while attempting a {text}. {text} records have been written to the output file.

Condition Identifier: AA\$ERROR_DURING_COPY Code: AA 1270

Description: The operation did not complete due to an error.

User Action: None required. If desired, correct the error and reenter the command.

--INFORMATIVE AA 1275-- {text} {text} {text} records from {text} to {text}. There were {text} trivial errors.

Condition Identifier: AA\$DISPLAY_COPY_RECORD_COUNT Code: AA 1275

Description: This message summarizes the effects of the command.

User Action: None required.

--WARNING AA 1280-- {text} detected that the input file {file} has its ruined flag set. The copy will be attempted, and, if successful, the ruined flag will be cleared in the output file.

Condition Identifier: AA\$COPYING_RUINED_FILE Code: AA 1280

Description: This message notifies the user that the input file specified on the command has its ruined flag set. Nevertheless, the file will be used if possible. The operation will not set the ruined flag for the output file.

User Action: None required. If the operation fails, the file data cannot be retrieved from this file. Request a reload of the last archived copy of the file and, if possible, update it to its state before it was ruined.

--INFORMATIVE AA 1285-- {text} deleted {text} records from nested file {text} in file {file}.

Condition Identifier: AA\$DISPLAY_DELETE_COUNT Code: AA 1285

Description: This message gives the number of records deleted from the nested file.

User Action: None required.

--WARNING AA 1290-- File {file} : this close was completed, but the file was not flushed because an exclusive file-freeze lock could not be obtained. Presumably another task will succeed in flushing before the last currently_active instance-of-open gives up the file. If not, the altered_not_closed flag will be left set.

Condition Identifier: AAE\$CANT_FLUSH_ON_CLOSE Code: AA 1290

Description: Although the close completed, it could not flush the file to disk because another instance of open has a lock on the file. Assuming the other instance of open closes successfully, flushing the file to disk, the file will remain usable. However, if the flush is not successful, the file will be left unusable, with its Altered_Not_Closed flag set.

User Action: Later, display the Altered_Not_Closed flag using the DISPLAY_KEYED_FILE_PROPERTIES command. If it is set, you should attempt to re-create the file using the COPY_KEYED_FILE command.

Further Information: The DISPLAY_KEYED_FILE_PROPERTIES and COPY_KEYED_FILE command descriptions are in the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1300-- The valid responses to this prompt are {text}. Please re-enter.

Condition Identifier: AAE\$IMPROPER_RESPONSE Code: AA 1300

Description: The response entered is invalid in this context.

User Action: Enter one of the listed responses.

--INFORMATIVE AA 1305-- As requested, this record was not deleted.

Condition Identifier: AAE\$THIS_RECORD_NOT_DELETED Code: AA 1305

Description: Informative message indicating that the user exercised the veto option to prevent deletion of the record.

User Action: None required. If desired, the record can be deleted later with another DELETE_RECORD command.

Further Information: For further information on the DELETE_RECORDS subcommand, see the NOS/VE Advanced File Management manual.

--WARNING AA 1310-- {text} could not {text} because nested file {text} is not located in {file}.

Condition Identifier: AAE\$NESTED_FILE_DOES_NOT_EXIST Code: AA 1310

Description: No nested file with the specified name exists in the file.

User Action: None required. To see the names of the nested files in the file, enter a DISPLAY_KEYED_FILE_PROPERTIES command.

Further Information: For more information on the nested-file subcommands and the DISPLAY_KEYED_FILE_PROPERTIES command, see the SCL Advanced File Management manual.

--WARNING AA 1315-- File {file} : There were no alternate_keys defined for nested file {text}. {text} will display only the ASCII and HEX representations of the records in the file.

Condition Identifier: AAE\$NO_ALTKEY_DEFS_TO_DISPLAY Code: AA 1315

Description: The display options requested a display of the alternate-key fields, but no alternate keys are defined for the nested file. Therefore, only the record data (in ASCII and hexadecimal representation) is displayed.

User Action: None required. Check that the correct nested file is selected. If appropriate, you can create alternate keys using the CREATE_ALTERNATE_INDEXES utility.

Further Information: For more information on DISPLAY_RECORDS and CREATE_ALTERNATE_INDEXES, see the NOS/VE Advanced File Management manual.

--WARNING AA 1320-- {text} was not processed because there were no records in nested_file {text} of file {file}, or the key value specified was greater than the largest key value in the file.

Condition Identifier: AA\$NO_RECORDS_TO_PROCESS Code: AA 1320

Description: No records were read for processing from the file for one of these reasons:

- The nested file is empty, or
- The specified key value is greater than the largest key value in the nested file.

User Action: None required. Check that the correct nested file is selected and that the correct key value was specified.

--WARNING AA 1325-- File {file} : The {text} file attribute is ignored when {text}.

Condition Identifier: AA\$FILE_ATTRIBUTE_IGNORED Code: AA 1325

Description: The file attribute specification is ignored in this context.

User Action: None required.

--WARNING AA 1330-- File {file} : A nested_file_name is not allowed when the file organization is sequential.

Condition Identifier: AA\$NFN_ILLEGAL_FOR_SEQ Code: AA 1330

Description: The file specification included a nested-file name, but the file is a sequential file, not a keyed file. Only keyed files can contain nested files; sequential files cannot.

User Action: Check that the correct file is specified. If so, remove the nested file name from the file specification.

Further Information: For more information on nested files, see the NOS/VE Advanced File Management or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--WARNING AA 1335-- File {file} : {text} was called but the key value supplied is not in the result set. The request did not change the contents of the result set file.

Condition Identifier: AA\$KEY_NOT_IN_RESULT_SET Code: AA 1335

Description: The primary-key value to be removed from the result set is not in the result set.

User Action: None required. Check that the correct key value was specified on the call.

Further Information: For more information on deleting key values from a result set, see the FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--WARNING AA 1340-- File {file} : {text} was called but the key value supplied is already in the result set. The request did not change the contents of the result set file.

Condition Identifier: AA\$KEY_FOUND_IN_RESULT_SET Code: AA 1340

Description: The primary-key value to be added to the result set is already in the result set.

User Action: None required. Check that the correct key value was specified on the call.

Further Information: For more information on adding values to a result set, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--INFORMATIVE AA 1345-- File {file} : {text} reached {text} in the result set file while skipping records.

Condition Identifier: AA\$RS_SKIP_TO_BOUNDARY Code: AA 1345

Description: Informative message notifying you that the result set file is now positioned at its beginning or end.

User Action: None required. If the result set file is positioned at its end, it must be repositioned before records can be read using the result set.

Further Information: For further information on the RSSKIP or RSREWIND calls and result sets in general, see the CYBIL Keyed-File & Sort/Merge Interfaces manual or the FORTRAN Usage manual.

--WARNING AA 1350-- {text} detected two or more records with the same primary key value in the input file(s). The records will be applied to the output file in the order in which they were read from the input file(s). The list of duplicate records has been saved in file {text}. Use DISPLAY_FILE rather than COPY_FILE to display {text} on a terminal if the output file is a direct_access file, or if the records may contain non_printable ASCII characters or binary or integer information.

Condition Identifier: AAES\$APPLYING_DUPLICATE_KEYS Code: AA 1350

Description: The command found two or more input records that have the same primary-key value. (If the primary key is collated, the values are the same after collation.) The command wrote the records with duplicate primary-key files to the file named in the message.

User Action: Check that the correct input files were specified. Look at the duplicate records to determine if the records are in error or the primary-key definition should be changed.

Further Information: The COPY_FILE and DISPLAY_FILE command descriptions are in the NOS/VE System Usage and NOS/VE Commands and Functions manuals. The attributes that define the primary key are described in the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1355-- File {file} : {text} will create an indexed_sequential file because the file_organization attribute was not specified.

Condition Identifier: AAES\$USING_DEFAULT_FILE_ORG Code: AA 1355

Description: The file_organization attribute was not specified for the new keyed file, so the command assumes indexed-sequential file organization.

User Action: None required. If the file was to use another keyed-file organization, it must be specified by the file_organization attribute.

Further Information: For further information on keyed-file organizations, see the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1360-- File {file} : {text} has deleted all of the records from {text}.

Condition Identifier: AAES\$NESTED_FILE_NOW_EMPTY Code: AA 1360

Description: Informative message notifying you that the DELETE_RECORDS subcommand has deleted all records from the currently selected nested file.

User Action: None required. The deleted records cannot be restored to the output file. The records still exist in the input file if it is not the same file as the output file.

Further Information: For more information on the DELETE_RECORDS subcommand, see the NOS/VE Advanced File Management manual.

--WARNING AA 1365-- Termination now will ruin file {text}. If this is what you want, type : RUIN FILE

Condition Identifier: AAES\$TERMINATE_WILL_RUIN_FILE Code: AA 1365

Description: Termination will ruin the file. If this is what you want, type: RUIN FILE.

User Action: None. This message is for users information.

--INFORMATIVE AA 1370-- The log_residence of the log for this file has changed from {text} to {text}.

Condition Identifier: AAES\$LOG_RESIDENCE_CHANGE Code: AA 1370

Description: The log residence has changed for this log.

User Action: None. Informative message.

--INFORMATIVE AA 1375-- Because the file path of this file has changed from {text} to {text}, a new log for this file has been created.

Condition Identifier: AAES\$FILE_PATH_CHANGE Code: AA 1375

Description: The file path for this file has changed. A new log for this file has been created.

User Action: None. Informative message only.

--INFORMATIVE AA 1376-- File {file} is no longer being logged.

Condition Identifier: AAE\$STOPPED_LOGGING Code: AA 1376

Description: The LOGGING_OPTIONS attribute for the specified file has been changed to NONE since the file was last opened, thereby stopping any further logging.

User Action: None. Informative Message.

Further Information: See the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1377-- File {file} : logging has been turned on, but will not actually happen until a BACPF has been done on the file.

Condition Identifier: AAE\$BEGIN_LOGGING Code: AA 1377

Description: Logging for the specified file has been turned on, that is, the LOGGING_OPTIONS attribute of the file has been changed to ENABLE_MEDIA_RECOVERY. However, before any log updates can be recorded, a backup of the file must be made using the Backup_Permanent_File (BACPF) utility.

User Action: Close the file, back up the file using BACPF, then resume.

Further Information: See the NOS/VE Advanced File Management manual.

--INFORMATIVE AA 1380-- Recover_Keyed_File is now attempting to restore {text} from the backup {text}.

Condition Identifier: AAE\$RESTORING_FILE_FROM_BACKUP Code: AA 1380

Description: RECOVER_KEYED_FILE is now attempting to restore from the backup.

User Action: None. Informative message.

--INFORMATIVE AA 1385-- Recover_Keyed_File is now applying the changes from the log to the file restored from the backup.

Condition Identifier: AAE\$APPLYING_CHANGES_FROM_LOG Code: AA 1385

Description: The Recover_Keyed_File is now applying the changes from the log to the file restored from the backup. Informative message.

User Action: None required. Informative message.

--WARNING AA 1390-- File {text} : AMP\$LOG_KEYED_FILE_BACKUP encountered an internal error: {text}. The file backup was not logged.

Condition Identifier: AAE\$INTERNAL_ERROR_IN_BACKUP Code: AA 1390

Description: The AMP\$LOG_KEYED_FILE_BACKUP encountered an internal error. The backup file was not logged.

User Action: Notify the Log Administrator at your site, to report an internal error.

--WARNING AA 1395-- File {text} : The backup for log entry dated {text} from {text} cannot be used because {text}.

Condition Identifier: AAE\$BACKUP_CANT_BE_USED Code: AA 1395

Description: The backup cannot be used for a particular log entry.

User Action: Check the backup being used.

--WARNING AA 1400-- File {text} : The log residence of the input file is {text}, but the log residence of the last valid backup in the system log ({text}) is {text}. The file will be restored using only the records from {text}.

Condition Identifier: AAE\$USER_LOG_RESIDENCE_IGNORED Code: AA 1400

Description: The log residence of the input file is ignored. The log residence of the last valid backup in the system log was ignored.

User Action: Notify the Log Administrator for assistance.

--WARNING AA 1405-- Because the library for Advanced_Access Methods is not available, backups cannot be logged for keyed files that have enable_media_recovery as a logging option.

Condition Identifier: AAE\$AAM_NOT_AVAILABLE Code: AA 1405

Description: The Advanced Access Methods library is not available. Backups are no logged for keyed files that have enable_media_recovery as a logging option.

User Action: AAM library must be made available.

--WARNING AA 1410-- IMPORTANT - File {text} : RECOVER_KEYED_FILE is deleting this file to ensure there is disk space for the backup file to be reloaded. To re-start RECOVER_KEYED_FILE, if it terminates before the file is successfully restored, you must supply the full file path, including the cycle number ({text}).

Condition Identifier: AAE\$DELETING_FILE_TO_RECOVER Code: AA 1410

Description: RECOVER_KEYED_FILE has deleted this file to make sure enough disk space is available for the backup file to be reloaded. If RECOVER_KEYED_FILE terminates before the file is restored, you must specify the full file path, including the cycle number when restarting the RECOVER_KEYED_FILE Utility.

User Action: None.

--WARNING AA 1415-- IMPORTANT - The LOG_RESIDENCE should also be set to {text} using the SET_FILE_ATTRIBUTES command if RECOVER_KEYED_FILE is restarted for this file.

Condition Identifier: AAE\$SETFA_FOR_LOG_RESIDENCE Code: AA 1415

Description: The LOG_RESIDENCE should also be set, using the SET_FILE_ATTRIBUTES command if the RECOVER_KEYED_FILE is restarted for this file.

User Action: Use the SET_FILE_ATTRIBUTES command to set LOG_RESIDENCE.

--INFORMATIVE AA 1420-- File {text}, which is required for the operation of the log, was missing from catalog {text}. It has been created.

Condition Identifier: AAE\$CREATING_LOG_FILE Code: AA 1420

Description: Required file is missing from the catalog. The file is created, if the CANCEL_LOG_CHANGES subcommand is entered.

User Action: None required. Informative error message.

--WARNING AA 1425-- {text} ignores the value for the {text} parameter when {text}.

Condition Identifier: AAE\$PARAM_VALUE_IGNORED Code: AA 1425

Description: Ignored the value for the parameter.

User Action: Verify that parameter is correct. May need to change parameter or delete the parameter.

Further Information: See the Administer_Recovery_Log description in the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1430-- Catalog {text} : {text} could not find backup entry {text}.

Condition Identifier: AAE\$BACKUP_ENTRY_DOES_NOT_EXIST Code: AA 1430

Description: The specified backup file could not be removed from the backup pool because it is not currently in the pool.

User Action: Check that the current file name or path is specified in REMOVE_FILE parameter.

Display the current backup pool by entering a DISPLAY_LOG_CONFIGURATION subcommand.

Further Information: See the CONFIGURE_LOG_BACKUP and DISPLAY_LOG_CONFIGURATION subcommand descriptions in the NOS/VE Advanced File Management Usage manual.

--WARNING AA 1435-- Catalog {text} : {text} could not find the directory for this catalog. The log could not be deleted.

Condition Identifier: AAE\$NO_DIRECTORY_FOR_CATALOG Code: AA 1435

Description: Could not find the directory for this catalog. The log could not be deleted.

User Action: Verify the correct directory for the catalog.

--INFORMATIVE AA 1440-- {text} has deleted catalog {text}.

Condition Identifier: AAE\$CATALOG_HAS_BEEN_DELETED Code: AA 1440

Description: The log has been deleted as requested.

User Action: None required.

--WARNING AA 1445-- File {file} : {text} is creating a direct_access or system_key file with primary key type=collated. Note however that key type has no effect unless file organization is indexed_sequential -- primary keys will be treated as if uncollated.

Condition Identifier: AAE\$COLLATED_IN_DA Code: AA 1445

Description: The file being created is a direct-access or system-key file with primary key type=collated. The key type has no effect, unless the file organization is indexed-sequential. Primary keys are treated as collated.

User Action: None required. Warning message only.

--WARNING AA 1450-- File {text} : {text} is waiting for the log control file to become available for update. A terminate break may be entered, but in that event the utility will terminate and no changes will be made to the log control file.

Condition Identifier: AAE\$WAITING_FOR_LOG_CNTL_FILE Code: AA 1450

Description: Administer_Recovery_Log was not able to attach the log control file for updating. This probably indicates that the log is in use or that someone else is modifying the log at the same time.

User Action: You can wait until the file is available or you can enter the terminate_break character (CDCNET default, %2 or CCP default, control_t) to abort the utility session. It is recommended that you modify a log only when no one else is using it.

--WARNING AA 1455-- Termination now will exit command utility {text} without performing any changes. If this is what you want, type: END_UTILITY

Condition Identifier: AAE\$TERMINATE_WILL_END_UTILITY Code: AA 1455

Description: Termination now will exit command utility without performing any changes. If this is what you want, type END_UTILITY.

User Action: None. This is just for user information.

--WARNING AA 1460-- Catalog {text} : {text} does not permit ({text}) as the performance level for parcels and ({text}) as the level for records. The latter must not be more reliable than the former, nor two steps less reliable. The less reliable one has been adjusted automatically to make an acceptable combination. You can reset them any time.

Condition Identifier: AAE\$PARCEL_PERF_GT_LOG_PERF Code: AA 1460

Description: If the performance level for parcels is SPEED, the performance level for records must also be SPEED. If the performance level for parcels is BALANCED, the performance level for records must be SPEED or BALANCED. If the performance level for parcels is RELIABILITY, the performance level specified for records can be SPEED, BALANCED, or RELIABILITY, but if RELIABILITY is specified, the actual performance level set is BALANCED.

User Action: Set the performance levels for records and parcels to one of the valid combinations.

Further Information: See the SET_PERFORMANCE_OPTION subcommand description in the NOS/VE Advanced File Management Usage manual.

--INFORMATIVE AA 1465-- File {text} : Recover_File_Media processed {text} records from the log. There were {text} trivial errors.

Condition Identifier: AAESUPDATES_APPLIED_FROM_LOG Code: AA 1465

Description: RECOVER_FILE_MEDIA processed records from the log. There were trivial errors.

User Action: None. This is an informative message only.

--INFORMATIVE AA 1470-- File {text} : Recover_File_Media found no update records in the log for this file.

Condition Identifier: AAESNO_UPDATES_APPLIED_FROM_LOG Code: AA 1470

Description: The RECOVER_FILE_MEDIA command found no update records in the log for this file.

User Action: None. Informative message.

--WARNING AA 1475-- File {text} : Recover_File_Media encountered a fatal error while attempting to update this file from the log.

Condition Identifier: AAESFATAL_ERROR_DURING_UPDATE Code: AA 1475

Description: the RECOVER_FILE_MEDIA command encountered a fatal error when attempting to update this file from the log.

User Action: Check the file being updated.

--WARNING AA 1480-- IMPORTANT - File {text} : The cycle number of this file is {text}.

Condition Identifier: AAESFILE_TO_RECOVER_CYCLE_NUM Code: AA 1480

Description: File to recover cycle number.

User Action: None.

--INFORMATIVE AA 1485-- In the future this problem may be avoided by increasing the life-span of the log or taking more frequent backups.

Condition Identifier: AAESLOG_TOO_SMALL Code: AA 1485

Description: This informative message may be avoided in the future by increasing the life-span of the log or taking more frequent backups.

User Action: None. Informative message.

--INFORMATIVE AA 1490-- Log {text} : No record was found in this log for a backup of file {text} prior to {text}.

Condition Identifier: AAESNO_BACKUP_RECORD_FOUND Code: AA 1490

Description: No backup record was found for this log.

User Action: Try another backup record.

--INFORMATIVE AA 1495-- Log {text} : Recover_File_Media is attempting to find a backup of file {text} in this log.

Condition Identifier: AAESSEARCHING_LOG_FOR_BACKUP Code: AA 1495

Description: RECOVER_FILE_MEDIA is attempting to find a backup of the file for this log.

User Action: None. Informative message.

--INFORMATIVE AA 1500-- Recover_File_Media has successfully reloaded file {text} using the backup record from log {text}. The date of the backup was {text}.

Condition Identifier: AAESUSING_BACKUP_FROM_LOG Code: AA 1500

Description: The RECOVER_FILE_MEDIA command has successfully reloaded the file using the backup record from the log. Includes the date of backup.

User Action: None. Informative message.

--WARNING AA 1505-- The {text} file attribute is ignored because this feature has not been implemented.

Condition Identifier: AA\$FILE_ATTR_UNIMPLEMENTED Code: AA 1505

Description: The file attribute has been ignored because the feature has not been implemented.

User Action: None.

--INFORMATIVE AA 1515-- {text} is attempting to mount tape {text} for verification.

Condition Identifier: AA\$MOUNTING_TAPE_FOR_VERIFY Code: AA 1515

Description: The tape you specified on CONFIGURE_LOG_BACKUP is being mounted and verified.

User Action: None. Informative message.

Further Information: See the CONFIGURE_LOG_BACKUP subcommand described in the NOS/VE Advanced File Management usage manual.

--INFORMATIVE AA 1520-- Tape {text} has been successfully validated and initialized.

Condition Identifier: AA\$TAPE_PASSED_VERIFICATION Code: AA 1520

Description: The tape you specified on CONFIGURE_LOG_BACKUP has been mounted, verified, and initialized successfully.

User Action: None. Informative message.

--WARNING AA 1525-- The {text} parameter value will be used. Note, however that the {text} feature is not currently implemented.

Condition Identifier: AA\$PARM_FUTURE_IMPLEMENTATION Code: AA 1525

Description: The parameter value you specified is for a feature not yet implemented. However, the information is stored and will be used when the feature is implemented.

User Action: If you want to use features that are currently implemented, specify a currently implemented parameter value.

Further Information: See the NOS/VE Advanced File Management usage manual.

--WARNING AA 1530-- Tape {text} currently has data on it.

Condition Identifier: AA\$TAPE_CONTAINS_DATA Code: AA 1530

Description: The tape you specified on CONFIGURE_LOG_BACKUP currently contains data. Following this message, you are asked whether you want to overwrite the tape or not.

User Action: If you want to overwrite the tape, the tape is overwritten with the new information. If not, the tape is not overwritten and you return to the Administer_Recovery_Log utility.

--INFORMATIVE AA 1535-- Tape {text} has been added to the list of log backups but has not been initialized.

Condition Identifier: AA\$TAPE_VERIFICATION_BYPASSED Code: AA 1535

Description: The tape file you specified on CONFIGURE_LOG_BACKUP has been added to the pool of backup files for the log. However, the tape containing the file has not been mounted and, therefore, not verified or initialized.

User Action: If you want the tape to be verified and initialized, specify VERIFY=TRUE on the CONFIGURE_LOG_BACKUP subcommand.

Further Information: See the CONFIGURE_LOB_BACKUP subcommand in the NOS/VE Advanced File Management usage manual.

--INFORMATIVE AA 1540-- File {file} : {text} has successfully discarded all previous log information for this file. Logging will resume after you have performed a BACPF of this file.

Condition Identifier: AAE\$VOID_OF_LOG_COMPLETED Code: AA 1540

Description: You have specified the VOID_LOG_FOR_RESTORED_FILE subcommand which has disassociated all logging done in the past for the specified keyed file. To start logging again, the specified keyed file must be backed up first using the Backup_Permanent_File (BACPF) utility.

User Action: To continue logging, back up the specified keyed-file using BACPF.

Further Information: See the NOS/VE Advanced File Management usage manual.

--WARNING AA 1545-- The information supplied for this command has been saved. The Configure_Log_Residence subcommand must also be entered for a new log.

Condition Identifier: AAE\$NEW_LOG_REQUIRES_CONLR Code: AA 1545

Description: The information that you supplied for the new log has been stored, however, it is meaningless until CONFIGURE_LOG_RESIDENCE has been entered. A new log is not created until it has been configured using CONFIGURE_LOG_RESIDENCE in the same session.

User Action: Make sure that you enter CONFIGURE_LOG_RESIDENCE during the current session before you quit. Otherwise, the new log and all its information is lost.

Further Information: See Creating a New Log in the NOS/VE Advanced File Management usage manual.

--WARNING AA 1550-- The Administer_Recovery_Log utility is exiting without configuring a new log.

Condition Identifier: AAE\$EXIT_WITH_NO_LOG_CONFIGURED Code: AA 1550

Description: A new log is not created until it is configured using CONFIGURE_LOG_RESIDENCE.

User Action: To create the log, specify USE_LOG following CONFIGURE_LOG_RESIDENCE.

Further Information: See Creating a New Log in the NOS/VE Advanced File Management usage manual.

--WARNING AA 1555-- Log {text} : This log catalog does not contain the log control file.

Condition Identifier: AAE\$LOG_WITH_NO_CONTROL_FILE Code: AA 1555

Description: The log catalog you deleted with DELETE_LOG does not contain a log control file.

User Action: None.

--WARNING AA 1560-- Log {text} : {text} cannot {text} the disk resident log backups for for this log, if any exist, because {text}.

Condition Identifier: AAE\$CANT_DELETE_LOG_BACKUPS Code: AA 1560

Description: The specified log catalog does not contain a log control file which means any disk resident log backups in the log catalog cannot be found to be deleted.

User Action: The disk resident log backups are now protected files, that is, they have attributes of ring 4,4,4. You may have to contact your site analyst to help you delete protected files.

--INFORMATIVE AA 1565-- {text} has deleted log {text}. Because the log catalog is not empty, it has not been deleted.

Condition Identifier: AAE\$LOG_HAS_BEEN_DELETED Code: AA 1565

Description: The specified log has been deleted. However, because the log catalog contained other user files, the log catalog has not been deleted.

User Action: None. Informative message.

--INFORMATIVE AA 1570-- {text} has successfully configured log {text}.

Condition Identifier: AAE\$LOG_SUCCESSFULLY_CONFIGURED Code: AA 1570

Description: This message is displayed when you terminate an Administer_Recovery_Log utility session in which a new log has been created and configured correctly.

User Action: None. Informative message.

--WARNING AA 1575-- The information supplied for this command has been saved. The {text} subcommand must also be entered {text}.

Condition Identifier: AAES\$SUBCOMMAND_INTERDEPENDENCE Code: AA 1575

Description: The information you supplied for this command has been stored. However, because this command is dependent on the specified subcommand, the information you supplied is meaningless until you specify the subcommand.

User Action: Enter the specified subcommand during the current session, otherwise, all information you supplied is lost when you quit.

Further Information: See the NOS/VE Advanced File Management usage manual.

--INFORMATIVE AA 1580-- {text} has been terminated at the request of the user.

Condition Identifier: AAES\$USER_REQUESTED_TERMINATION Code: AA 1580

Description: The specified subcommand has been terminated by a terminate break character. You are returned to the Administer_Recovery_Log utility.

User Action: Before proceeding, you should specify DISPLAY_LOG_CONFIGURATION to display what options are currently in effect.

--WARNING AA 1585-- {text} is unable to validate the information supplied for the log backup account because the following error occurred:

Condition Identifier: AAES\$UNABLE_TO_TEST_SETLBA_PARMS Code: AA 1585

Description: To test the validation of the user and password accounting information, a batch job is executed. In this case, the test failed due to the displayed information.

User Action: Specify the correct user and password information on the SET_LOG_BACKUP_ACCOUNT subcommand.

--WARNING AA 1590-- {text} has executed a test job to validate the log backup account information provided. The job returned the following status:

Condition Identifier: AAES\$BAD_LOG_BKUP_INFO Code: AA 1590

Description: To test the validation of the user and password accounting information, a batch job is executed. In this case, the test failed due to the displayed information.

User Action: You must delete the log and recreate the log with the correct user-name and password on the SET_LOG_BACKUP_ACCOUNT subcommand.

--INFORMATIVE AA 1595-- Modifications have been lost on file {file}. {text} will now try to reset the RESPF_MODIFICATION_MISMATCH condition.

Condition Identifier: AAES\$MODIFICATIONS_LOST_ON_FILE Code: AA 1595

Description: The log has been restored by the Restore_Permanent_File utility. As a result, an earlier version of the log has been restored.

User Action: Informative message. The Restore_log utility will attempt to clear the condition.

--INFORMATIVE AA 1600-- The attempt to {text} was successful.

Condition Identifier: AAES\$ATTEMPT_WAS_SUCCESSFUL Code: AA 1600

Description: The specified action was completed successfully.

User Action: None. Informative message.

--INFORMATIVE AA 1605-- The attempt to {text} was unsuccessful. Try {text}.

Condition Identifier: AAES\$MAKE_ANOTHER_ATTEMPT Code: AA 1605

Description: The log control file cannot be restored from the most recently written backup file.

User Action: Enter the RESTORE_LOG_CONTROL_FILE subcommand again, but specify the next most recent backup file.

--INFORMATIVE AA 1610-- The ENABLE_LOG subcommand is unnecessary. The DAMAGED flag has not been set.

Condition Identifier: AA\$ENABLE_LOG_UNNECESSARY Code: AA 1610

Description: The log is already enabled.

User Action: Do not use the ENABLE_LOG subcommand.

--INFORMATIVE AA 1615-- Enter ENABLE_LOG to clear the DAMAGED condition.

Condition Identifier: AA\$LOG_READY_TO_BE_ENABLED Code: AA 1615

Description: The damaged repositories have been replaced successfully from the backup files. The log is now ready to be enabled and made available for general use.

User Action: Use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log.

--INFORMATIVE AA 1625-- {text} completed successfully.

Condition Identifier: AA\$SUBCOMMAND_SUCCESSFUL Code: AA 1625

Description: The RESTORE_REPOSITORIES subcommand of the Restore_Log utility completed successfully. However, check prior messages that may indicate the extent of damage to the log, if any.

User Action: None. Informative message.

--INFORMATIVE AA 1630-- No problems were found during the validation of the log.

Condition Identifier: AA\$LOG_PASSED_VALIDATION Code: AA 1630

Description: No damage to the log was detected by the VALIDATE_LOG subcommand; the log remains enabled.

User Action: None. Informative message.

--INFORMATIVE AA 1635-- No plausible date-time seen at end of any repository.

Condition Identifier: AA\$NO_PLAUSIBLE_DATE_TIME Code: AA 1635

Description: There are no log entries in any of the repositories.

User Action: If you want to restore repositories from the backup files, delete the repositories using the DELETE_REPOSITORIES subcommand of the Restore_Log utility, then use the RESTORE_REPOSITORIES subcommand.

--INFORMATIVE AA 1640-- The last available update found in the log was at {text} {text}. Please ensure this is correct before attempting to use this log for recovery or resuming the logging of updates.

Condition Identifier: AA\$CHECK_TIME_DATE_OF_LOG Code: AA 1640

Description: The specified date and time indicate the last update in the most recent repository. Thus, you can restore a damaged keyed file up to this time using the Recover_Keyed_File utility.

User Action: None. Informative message.

--WARNING AA 1645-- PMP\$GET_COMPACT_DATE_TIME or PMP\$COMPUTE_DATE_TIME returned the following bad status, and may have left random data in the repository_expiration_time in the log control file.

Condition Identifier: AA\$POSSIBLE_BAD_TIME_IN_LOG Code: AA 1645

Description: The date and/or time for the log control record of the log control file could not be obtained from the system.

User Action: Be aware that the date and time in the log control record may be missing or unreadable. The log is available only for recovering keyed files. To begin recording updates again, you must switch to a different log, or you must delete the log, then recreate it.

--WARNING AA 1650-- {text} was successful but {text}.

Condition Identifier: AAE\$CONDITIONAL_SUCCESS Code: AA 1650

Description: The log has been successfully restored, however, the latest updates in the active repository may be lost.

User Action: The log is available only for recovering keyed files. To begin recording updates again, you must switch to a different log, or you must delete the log, then recreate it.

--INFORMATIVE AA 1655-- Logging may be resumed on log {text}.

Condition Identifier: AAE\$LOGGING_MAY_BE_RESUMED Code: AA 1655

Description: The log has been restored successfully. It is available for recovery purposes and also for recording updates.

User Action: None. Informative message.

--WARNING AA 1660-- Log {text}: This log may only be used to attempt recovery of keyed files. Logging may not be resumed on this log.

Condition Identifier: AAE\$LOG_USABLE_FOR_RECOVERY Code: AA 1660

Description: The log has been restored, however, because the log control file and/or the active repository was replaced, the log is available only for recovering keyed files.

User Action: If you want to continue recording updates, you must switch to a different log, or you must delete the log, then recreate it.

--WARNING AA 1665-- Log {text}: This log is ready to be enabled. To enable the log use the Enable_Log subcommand of Restore_Log.

Condition Identifier: AAE\$LOG_NOT_YET_ENABLED Code: AA 1665

Description: The log is ready to be enabled.

User Action: Use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log.

--WARNING AA 1670-- Log {text}: The Restore_Log utility is terminating with the log disabled.

Condition Identifier: AAE\$LOG_CURRENTLY_DISABLED Code: AA 1670

Description: You have exited the Restore_Log utility without enabling the indicated log.

User Action: Be aware that the log is disabled and cannot be used until it is enabled.

--INFORMATIVE AA 1675-- The Restore_Repository subcommand should now be entered.

Condition Identifier: AAE\$READY_TO_RESTORE_REPOSITORY Code: AA 1675

Description: The log control file has been successfully restored or replaced.

User Action: Use the RESTORE_REPOSITORIES subcommand of the Restore_Log utility to restore the damaged repositories.

--WARNING AA 1680-- Because {text}, the entries in the currently active repository, if any, have been lost.

Condition Identifier: AAE\$ACTIVE_REPOSITORY_LOST Code: AA 1680

Description: The log updates in the active repository have been lost.

User Action: Because the updates in the active repository have been lost, the log is available only for recovering keyed files. To begin recording updates again, you must switch to a different log, or you must delete the log, then recreate it.

--INFORMATIVE AA 1685-- The log is ready to be enabled by entering the Enable_Log subcommand of Restore_Log.

Condition Identifier: AAE\$READY_TO_ENABLE_LOG Code: AA 1685

Description: The RESTORE_REPOSITORIES subcommand has completed successfully; all damaged repositories have been repaired or replaced from the backup files. The log is now ready to be enabled.

User Action: Use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log.

--WARNING AA 1690-- The log may be enabled by entering the Enable_Log subcommand of Restore_Log. The log may then only be used for recovering keyed files with Recover_Keyed_File. Logging may not resume on this log.

Condition Identifier: AAE\$READY_TO_ENABLE_FOR_RECKF Code: AA 1690

Description: The log has been restored and is now ready to be enabled; that is, made available for general use. However, recovery information in the log has been lost. As a result, the log is available only for recovering keyed files.

User Action: Before using the log to recover keyed files, use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log. To continue recording updates, you must switch to a different log, or you must delete the log, then recreate it.

--WARNING AA 1695-- Log {text} was not available or is damaged. Recover_File_Media will not be able to apply the updates to the keyed file from the log.

Condition Identifier: AAE\$USER_LOG_NOT_AVAILABLE Code: AA 1695

Description: The indicated log is either currently in use or is damaged.

User Action: If the log is currently in use, try again later. If the log is damaged, use the Restore_Log utility to repair it.

--WARNING AA 1700-- {text} has reinitialized log file {text}. If there were any entries in this repository, they have been lost.

Condition Identifier: AAE\$REPOSITORY_REINITIALIZED Code: AA 1700

Description: The indicated log has been restored. However, all updates in the active repository have been lost. As a result, the log will be available only for recovering keyed files; no further updates can be recorded on this log.

User Action: To continue recording updates, you must switch to a different log, or you must delete the log, then recreate it.

--INFORMATIVE AA 1705-- Log file {text} is damaged. {text} will attempt to delete the file and recreate it. {text}

Condition Identifier: AAE\$RECREATING_MINOR_FILE Code: AA 1705

Description: An attempt was made to restore the minor log files (the tape ownership file or the log journal), but failed. An attempt to delete the file, then recreate it is made.

User Action: None. Informative message.

--WARNING AA 1710-- The backup found in the system log cannot be used because the system log record has {text} for the log residence and the keyed file attribute value is {text}.

Condition Identifier: AAE\$LOG_RESIDENCE_MISMATCH Code: AA 1710

Description: The LOG_RESIDENCE of the keyed file does not match the LOG_RESIDENCE of the keyed file backup.

User Action: Specify the correct LOG_RESIDENCE using the SET_FILE_ATTRIBUTES command.

--WARNING AA 1715-- {text} was unable to set the permits for file {text}. Please have the owner set the permits of this file to: {text}.

Condition Identifier: AAE\$UTIL_CANT_SET_PERMITS Code: AA 1715

Description: You can create the lock file, but only the owner of the master catalog can set the file permit for the lock file.

User Action: The owner must set the specified permits for the lock file.

--ERROR AA 2000-- File{text} {file}{text} : does not contain the key used in this {text} operation-- {text}.

Condition Identifier: AA\$KEY_NOT_FOUND Code: AA 2000

Description: The operation specified a key value that is not in the file.

If the key value does not appear in the error message, it is on the error listing file (\$ERRORS) if the message_control attribute requested listing of trivial (nonfatal) errors.

User Action: Check that the correct key value was specified. If alternate keys are used, check that the correct alternate key was selected.

Further Information: Specification of key values is described in the COBOL Usage manual, the FORTRAN Language Definition Usage manual, and the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2005-- File{text} {file}{text} already contains a record with the same key that this {text} operation uses. -- {text}.

Condition Identifier: AA\$KEY_ALREADY_EXISTS Code: AA 2005

Description: The primary-key value used in the specified operation already exists in the file. Primary-key values must be unique.

If the primary key is a collated key, the key values must be unique after collation.

If the key value does not appear in the error message, it is on file \$ERRORS if the message_control attribute requested listing of trivial (nonfatal) errors.

User Action: Ensure that the primary-key value of each record is unique. If appropriate, issue a replace request to replace the existing record with the new record.

Further Information: To read about replacing the record, see the FORTRAN Language Definition Usage, COBOL Usage, or CYBIL Keyed-File and Sort/Merge Interfaces Usage manual. To read about how collation works, see the Collation Tables appendix in the NOS/VE Advanced File Management manual.

--ERROR AA 2010-- File{text} {file}{text} : {text} found a record longer than the working storage area. The last part of record is not delivered to the working storage area. {text}

Condition Identifier: AA\$RECORD_LONGER_THAN_WSA Code: AA 2010

Description: The working storage area is not as long as the record that was to be read. However, the working storage area does contain the first part of the record.

If the key value does not appear in the error message and the message_control attribute requested listing of trivial errors, the key value is on file \$ERRORS.

User Action: Increase the length of the working storage area.

Further Information: Reading keyed files is described in the FORTRAN Usage, COBOL Usage, and the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2015-- File{text} {file}{text} : {text} is not done because the file already contains the maximum allowed number of records. To increase the limit, close the file and re-open with a higher value for the RECORD_LIMIT attribute. {text}

Condition Identifier: AA\$FILE_AT_USER_RECORD_LIMIT Code: AA 2015

Description: The record could not be written because its addition to the file would exceed the record limit specified by the record_limit attribute value.

If the key value does not appear in the error message, it is on file \$ERRORS if the message_control attribute requested listing of trivial errors.

User Action: To see the current record_limit attribute value, enter a DISPLAY_FILE_ATTRIBUTES command. To increase the record limit, increase the record_limit attribute value with a CHANGE_FILE_ATTRIBUTE command.

Further Information: See the descriptions of the DISPLAY_FILE_ATTRIBUTES and CHANGE_FILE_ATTRIBUTE commands in the NOS/VE System Usage manual.

--ERROR AA 2020-- File{text} {file}{text} : This indexed-sequential file now has the maximum (15) index levels and the primary index block 0 is almost full so {text} is denied because the operation could require creation of another index level. The file should be re-created with a smaller INDEX_BLOCK_PADDING attribute and/or a larger block size. It is possible that just copying to a new INDEXED_SEQUENTIAL file with same-size blocks would solve the problem, if it is due to strange patterns of deleting and putting of records in the past. {text}

Condition Identifier: AAE\$FILE_FULL_NO_PUTS_OR_REPS Code: AA 2020

Description: No more records can be written to the file because the file now has 15 index levels and the primary index block is nearly full.

User Action: To write more records to the file, re-create the file with fewer index levels. The new file may require a smaller index_padding percentage and/or larger blocks.

Further Information: The NOS/VE Advanced File Management Usage manual describes how to re-create an indexed-sequential file.

--ERROR AA 2035-- File{text} {file}{text} : operation {text} has positioned the file at the key value but cannot complete because this or another instance-of-open of this task has the key value locked. {text}

Condition Identifier: AAE\$PRIMARY_KEY_LOCKED Code: AA 2035

Description: Although the request positioned the file at the specified key value, it cannot be perform the requested operation on the key value because a conflicting lock exists for the key value.

User Action: Try again later. The program could request a lock on the key value and specify waiting for the lock. The request would then succeed unless the wait exceeds the timeout period.

Or, before processing the file, you could attach it for exclusive access (share_modes=none) so that the file would not be shared while you use it.

Further Information: For further information, refer to the Keyed-File Sharing discussions in the COBOL Usage, FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--ERROR AA 2040-- File{text} {file}{text} : the request by operation {text} for a lock (with WAIT) on a primary-key value (or the file) was denied because a deadlock was detected. {text}

Condition Identifier: AAE\$KEY_DEADLOCK Code: AA 2040

Description: The lock cannot be granted because a deadlock situation exists. The instance of open holding the lock is waiting for a lock held by another instance of open. That lock cannot be granted until this instance of open releases a lock it holds.

User Action: Change the program so that, when it receives this status, it releases all locks it holds to clear the deadlock situation. To prevent a deadlock when an operation requires several locks, request a lock on the entire file.

Further Information: See the UNLOCKK and LOCKF call descriptions in the FORTRAN Usage manual or the AMP\$UNLOCK_KEY and AMP\$LOCK_FILE call descriptions in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2045-- File{text} {file}{text} : the request by operation {text} for a lock (with WAIT) on a primary-key value (or the file) was denied because self-deadlock was detected. This means this task already has the key value locked or the whole file locked. {text}

Condition Identifier: AAE\$KEY_SELF_DEADLOCK Code: AA 2045

Description: The task already has the requested lock. It may be that the task already has a file lock (a lock on all key values in the nested file), and is now requesting a lock on a key value.

User Action: Correct the program logic so that it cannot request locks that it already has. When a lock is requested without automatic unlock, the program may be required to explicitly clear the lock.

Further Information: For more information on locking, see the COBOL Usage, FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--ERROR AA 2050-- File{text} {file}{text} : operation {text} failed because this operation is disallowed inside a parcel. {text}

Condition Identifier: AAESCANT_UNLOCK_KEY_IN_PARCEL Code: AA 2050

Description: This operation is not allowed within a parcel.

User Action: None.

--ERROR AA 2055-- File{text} {file}{text} : operation {text} requests a lock with wait, but the wait time limit ran out before the existing conflicting lock was released. {text}

Condition Identifier: AAESKEY_TIMEOUT Code: AA 2055

Description: The system could not perform the operation because the wait time limit expired before the conflicting lock was released. The default wait time limit is 60 seconds; however, it can be changed using the SCL variable AAV\$RESOLVE_TIME_LIMIT.

User Action: Try again later.

If appropriate, display the value of the AAV\$RESOLVE_TIME_LIMIT variable, if it exists. A new value can be assigned to the variable by an SCL statement. The SCL statement can be executed from within a program. For example, the following statement can change the value from within a FORTRAN program:

```
Call SCLCMD('create_variable, name=AAV$RESOLVE_TIME_LIMIT,
+          kind=integer, value=45, scope=local')
```

Similarly, the following call can change the value from within a CYBIL program:

```
CLP$SCAN_COMMAND_LINE('create_variable,
                        name=AAV$RESOLVE_TIME_LIMIT, kind=integer,
                        value=45, scope=local', status);
```

--ERROR AA 2060-- File{text} {file}{text} : operation {text} failed because part of the file remained locked for longer than the wait time limit. {text}

Condition Identifier: AAESCANT_LOCK_RASP_FOR_OP Code: AA 2060

Description: The system could not perform the operation because the wait time limit expired before it could get access to part of the file. The default wait time limit is 60 seconds; however, it can be changed using the SCL variable AAV\$RESOLVE_TIME_LIMIT.

User Action: Try again later. If possible, attach the file for exclusive access (share_modes=none).

If appropriate, display the value of the AAV\$RESOLVE_TIME_LIMIT variable, if it exists. A new value can be assigned to the variable by an SCL statement. The SCL statement can be executed from within the program. For example, the following statement can change the value from within a FORTRAN program:

```
Call SCLCMD('create_variable, name=AAV$RESOLVE_TIME_LIMIT,
+          kind=integer, value=45, scope=local')
```

Similarly, the following call can change the value from within a CYBIL program:

```
CLP$SCAN_COMMAND_LINE('create_variable,
                        name=AAV$RESOLVE_TIME_LIMIT, kind=integer,
                        value=45, scope=local', status);
```

--ERROR AA 2065-- File{text} {file}{text} : operation {text} failed because a block remained locked for longer than the wait time limit. {text}

Condition Identifier: AAESCANT_LOCK_BLOCK Code: AA 2065

Description: The system could not perform the operation because the wait time limit expired before it could get access to part of the file. The default wait time limit is 60 seconds; however, it can be changed using the SCL variable AAV\$RESOLVE_TIME_LIMIT.

User Action: Try again later. If possible, attach the file for exclusive access (share_modes=none).

If appropriate, display the value of the AAV\$RESOLVE_TIME_LIMIT variable, if it exists. A new value can be assigned to the variable by an SCL statement. The SCL statement can be executed from within the program. For example, the following statement can change the value from within a FORTRAN program:

```
Call SCLCMD('create_variable, name=AAV$RESOLVE_TIME_LIMIT,
+          kind=integer, value=45, scope=local')
```

Similarly, the following call can change the value from within a CYBIL program:

```
CLP$SCAN_COMMAND_LINE('create_variable,
                        name=AAV$RESOLVE_TIME_LIMIT, kind=integer,
                        value=45, scope=local', status);
```

--ERROR AA 2070-- File{text} {file}{text} : operation {text} failed because this operation cannot be performed on a SYSTEM_KEY file unless the key value belongs to an existing record. {text}

Condition Identifier: AAESCANT_LOCK_UNUSED_KEY Code: AA 2070

Description: This file operation cannot be performed on a SYSTEM_KEY file unless the key value belongs to an existing record.

User Action: Identify key value on existing record.

--ERROR AA 2075-- File{text} {file}{text} : operation {text} failed because another instance-of-open has the primary-key value (or the file) locked, or this instance-of-open has a lock already and is now asking for a conflicting lock. Note that any existing PRESERVE_CONTENT lock on the primary-key value must be cleared before the primary-key value can be locked for updating.

Condition Identifier: AAESKEY_FOUND_LOCK_NO_WAIT Code: AA 2075

Description: The lock request did not specify waiting so it failed immediately when a conflicting lock was found.

User Action: Try again requesting wait.

Further Information: See the lock call descriptions in the FORTRAN Usage and CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2080-- File{text} {file}{text} : operation {text} failed because the same instance-of-open already has the primary-key value (or the file) locked in an incompatible way. {text}

Condition Identifier: AAESKEY_ALREADY_LOCKED Code: AA 2080

Description: Either the instance of open has a Preserve_Content lock and has requested either a Preserve_Access_and_Content or Exclusive_Access lock or, vice versa, it has a Preserve_Access_and_Content or Exclusive_Access lock and has requested a Preserve_Content lock.

User Action: Clear the incompatible lock before requesting a new lock. If appropriate, request a Preserve_Access_and_Content lock when reading the record and change it to an Exclusive_Access lock if the record is to be changed.

Further Information: Lock processing is described in the keyed-file interface section of the FORTRAN Usage and CYBIL Keyed-File and Sort/Merge Interfaces manuals.

--ERROR AA 2085-- File{text} {file}{text} : operation {text} failed because, the file is not attached exclusively, neither the primary key nor the file has been locked, and you have set the prelocking-required flag. The key or the file must be locked with Exclusive_Access or Preserve_Access_and_Content intent before the record can be deleted or replaced. {text}

Condition Identifier: AA\$KEY_NOT_ALREADY_LOCKED Code: AA 2085

Description: The operation failed because the file could not be shared and the instance of open does not have an Exclusive_Access or Preserve_Access_and_Content lock on the primary-key value of the record to be deleted or replaced.

User Action: Either request the required lock before attempting to delete or replace the record or attach the file for exclusive access (share_modes=none) before updating the file. (Locks are not required when the file cannot be shared.)

Further Information: See the keyed-file sharing discussion in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual. See the ATTACH_FILE command description in the SCL System Interface or SCL Quick Reference manual.

--ERROR AA 2090-- File{text} {file}{text} : operation {text} failed because the primary-key value (or the file) was not locked by this instance-of-open. {text}

Condition Identifier: AA\$KEY_NOT_PREVIOUSLY_LOCKED Code: AA 2090

Description: The operation requires an Exclusive_Access lock on the primary-key value of the record to be deleted or replaced.

User Action: Attach the file for exclusive access (share_modes=none) or change the program so that it requests an Exclusive_Access lock on the primary-key value before it attempts the delete or replace.

Further Information: Locking is described in the CYBIL Keyed-File and Sort/Merge Interfaces manual, the FORTRAN Usage manual, and the COBOL Usage manual.

--ERROR AA 2095-- File{text} {file}{text} : operation {text} failed because a DIRECT_ACCESS or SYSTEM_KEY file cannot be positioned (by AMP\$START) using a primary-key value. {text}

Condition Identifier: AA\$NO_DA_OR_SK_START Code: AA 2095

Description: It is not appropriate to position a direct-access or system-key file to a specific key value in the file because reading records from that position forward returns records randomly, not in sorted order as for an indexed-sequential file.

User Action: To read a single record by its key value, perform a random-access read (AMP\$GET_KEY or AMP\$GET_LOCK_KEY). To read all records in a direct-access or system-key file, rewind the file to its beginning and read through the entire file sequentially without intervening update operations.

Further Information: For further information on reading a direct-access file, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--ERROR AA 2100-- File{text} {file}{text} : operation {text} failed because the new record contains an alternate key value that already exists in another record for an alternate key that does not allow duplicate values. {text}

Condition Identifier: AA\$DUPLICATE_ALTERNATE_KEY Code: AA 2100

Description: The new record could not be added to the file because its alternate-key value matches an alternate-key value already in the file, and the alternate key does not allow duplicates.

User Action: If appropriate, replace the existing record with the new record having the same alternate-key value or delete and redefine the alternate key to allow duplicates.

Further Information: To read about duplicate key control when writing records, see the COBOL Usage manual (DUPLICATES phrase) or the FORTRAN Language Definition Usage manual (PUT and REPLC calls) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2105-- File{text} {file}{text} : {text} found a sparse key control character which indicated key inclusion, but the record was too short to contain the key. The operation continues, and the key is excluded. {text}

Condition Identifier: AAE\$SPARSE_KEY_BEYOND_EOR_FIRST Code: AA 2105

Description: Although the character in the sparse key control position indicates that the alternate-key value for the record should be included in the alternate index, it is not included because the alternate-key field extends beyond the end of the record.

User Action: None required. However, you should determine why the record did not contain a complete alternate-key value.

Further Information: To read about sparse-key control and apply error processing, see the NOS/VE Advanced File Management manual or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2110-- File{text} {file}{text} : {text} could not find a key delimiter character for a variable-length alternate key. The operation continues, with no alternate key value taken from the record. {text}

Condition Identifier: AAE\$MSG_KEY_DELIMITER_FIRST Code: AA 2110

Description: While creating an alternate index for a variable-length alternate key, a key value was found that is longer than the key length specified for the key. This was determined while collecting key values from a record when no delimiter character was found for key_length+1 bytes.

User Action: Determine whether the contents of the record or the alternate-key definition is in error. Correct the error and reattempt the operation.

Further Information: Variable-length alternate keys and alternate-key creation are described in the CYBIL Keyed-File & Sort/Merge Interfaces manual, the FORTRAN Usage manual, and the NOS/VE Advanced File Management manual.

--ERROR AA 2115-- File{text} {file}{text} : {text} is not allowed because the instance-of-open already holds at least 1024 key locks.

Condition Identifier: AAE\$TOO_MANY_KEYLOCKS Code: AA 2115

Description: The instance of open cannot be granted another lock until it releases one or more of the locks it already holds.

User Action: Either change the program so that it releases locks (either automatically or explicitly) or, if it requires concurrent locks on several key values, it should request a file lock.

Further Information: See the UNLOCKK and LOCKF call descriptions and the \$AUTOMATIC_UNLOCK FIT value description in the FORTRAN Usage manual. See the AMP\$UNLOCK_KEY and AMP\$LOCK_FILE call descriptions in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2120-- File{text} {file}{text} : {text} is not allowed because the LOCK_INTENT indicates an intent to update the file, but the set of ACCESS_MODES does not include append or shorten permission. Without append or shorten access, the only valid LOCK_INTENT is PRESERVE_CONTENT.

Condition Identifier: AAE\$NO_EXCLUSIVE_IF_READ_ONLY Code: AA 2120

Description: A request that specified Preserve_Access_and_Content or Exclusive_Access lock_intent is denied if the instance-of-open does not have append or shorten access. Those lock_intent values indicate that the program intends to write, replace, or delete a record, which it cannot do without append or delete access.

User Action: Either change the specified lock_intent to Preserve_Content or attach and open the file for append and shorten access.

Further Information: To read about lock intents and the OPENM call, see the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual. To read about the ATTACH_FILE command, see the NOS/VE System Usage manual.

--ERROR AA 2125-- File{text} {file}{text} : {text} failed because UNLOCK_CONTROL is set to AMC\$AUTOMATIC, which caused the key to be automatically unlocked at the start of this request. AMP\$UNLOCK_KEY and UNLOCK_CONTROL = AMC\$AUTOMATIC are alternative methods of unlocking keys and should not be used together. {text}

Condition Identifier: AAE\$INCOMPATIBLE_UNLOCKING Code: AA 2125

Description: An unlock request was issued for a lock whose lock request specified automatic unlocking. The lock was automatically unlocked when the unlock call was issued; the unlock operation then found no lock on the key value.

User Action: Remove either the automatic unlock specification from the lock request or the explicit unlock call.

Further Information: For a discussion of lock clearing, see the CYBIL Keyed-File & Sort/Merge Interfaces manual or the FORTRAN Usage manual.

--ERROR AA 2130-- File {file} : {text} was called with key relation of AMC\$EQUAL_KEY, but an exact match of the key value or major key value could not be found.

Condition Identifier: AAE\$KEY_NOT_LOCATED Code: AA 2130

Description: The request specified that the key value specified on the request must exactly match the entire key value or the beginning of the key value in the record. Such a record does not exist in the currently selected nested file.

User Action: Check that the correct key value was specified on the request. If appropriate, change the key_ relation so that it asks for the first record whose key value is greater than or equal to the specified key value.

Further Information: For more information on positioning and reading a keyed file, see the CYBIL Keyed-File & Sort/Merge Interfaces manual, the COBOL Usage manual, or the FORTRAN Usage manual.

--ERROR AA 2300-- Recovery Log file {text} is damaged; please notify your Recovery Log Administrator, who should restore the log control file with the Restore_Log command utility.

Condition Identifier: AAE\$ADMIN_MUST_RESTORE_LCF Code: AA 2300

Description: An attempt to read from or write to the log has occurred, however, the log control file is damaged.

User Action: It is recommended that the log administrator restore the log control file using the Restore_Log utility.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2301-- Recovery Log file {file} is damaged; please notify your Recovery Log Administrator, who should run the Restore_Log command utility.

Condition Identifier: AAE\$ADMIN_MUST_RESTORE_LOG Code: AA 2301

Description: An attempt to read from or write to the log has occurred, however, one of the repositories is damaged.

User Action: It is recommended that the log administrator restore the log repository using the Restore_Log utility.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2302-- File {text} : The contents of this log control file are invalid because {text}.

Condition Identifier: AAE\$BAD_LOG_CNTL_FILE_CONTENTS Code: AA 2302

Description: Contents of log file are invalid.

User Action: Check contents of log file.

--ERROR AA 2303-- Catalog {text} : The log residence may not be in the \$LOCAL catalog.

Condition Identifier: AAE\$CANNOT_USE_\$LOCAL_CATALOG Code: AA 2303

Description: A log cannot be configured in the temporary file catalog, \$LOCAL.

User Action: Specify a permanent file catalog in which the log is to be configured.

--ERROR AA 2304-- Recovery Log {text} has been restored after incurring damage; it is only available for recovering lost updates by means of the Recover_Keyed_File utility. If your keyed file is okay or has been recovered and you wish to resume update operations on it, consult your Recovery Log Administrator, who must either delete and recreate this log or change you over to a new log.

Condition Identifier: AAE\$CANT_WRITE_ON_RESTORED_LOG Code: AA 2304

Description: The log has been restored, however, with some potential loss of log information. As a result, no further logging can occur.

User Action: Consult your log administrator to either delete and recreate the log or move you to a new log.

--ERROR AA 2305-- Catalog {text} : A file cycles reference is not permitted on a catalog path.

Condition Identifier: AAE\$CATALOG_MAY_NOT_HAVE_CYCLE Code: AA 2305

Description: A catalog path cannot end with a cycle reference (an integer or one of the cycle indicators, \$NEXT, \$HIGH, or \$LOW).

User Action: Check that a catalog path, not a file path is specified.

--ERROR AA 2306-- Catalog {text} : The log residence may not exceed 512 characters.

Condition Identifier: AAE\$CATALOG_PATH_TOO_LONG Code: AA 2306

Description: The specified catalog path is longer than 512 characters.

User Action: Correct the specified catalog path so it is no longer than 512 characters.

--ERROR AA 2307-- The log is temporarily full, pending expiration of Switch_Suppression_Time. The Log Administrator should be notified of this condition, as it probably indicates a log configuration problem.

Condition Identifier: AAE\$LOG_TEMPORARILY_FULL Code: AA 2307

Description: the log is temporarily full, pending the expiration of SWITCH_SUPPRESSION_TIME. The log administrator should be notified of this condition. May indicate a log configuration problem.

User Action: Notify the log administrator.

--ERROR AA 2308-- Recovery Log {text} is being restored by your Recovery Log Administrator; please try again later.

Condition Identifier: AAE\$RESTORE_LOG_IN_PROGRESS Code: AA 2308

Description: You are trying to open a keyed-file, however, the specified log is currently being restored. The keyed-file cannot be opened until the log restoration is complete.

User Action: Wait until the log has been restored before opening the keyed file.

--ERROR AA 2309-- {text} cannot be done because the user ring is beyond the log ring.

Condition Identifier: AAE\$USER_RING_BEYOND_LOG_RING Code: AA 2309

Description: Operation cannot be done because the user ring is beyond the log ring.

User Action: Notify the Log Administrator for assistance.

--ERROR AA 2310-- {text} cannot be done because the log is not opened for write permission.

Condition Identifier: AAE\$LOG_NOT_OPENED_FOR_WRITE Code: AA 2310

Description: Operation cannot be done, log is not opened for WRITE permission.

User Action: Open log for WRITE permission.

--ERROR AA 2311-- {text} cannot be done until TSP\$POSITION_LOG_BY_TIME or TSP\$POSITION_LOG_BY_ADDRESS has established a log position for this instance of open.

Condition Identifier: AAE\$LOG_NOT_POSITIONED_FOR_GETN Code: AA 2311

Description: Operation cannot be done until TSP\$POSITION_LOG_BY_TIME or TSP\$POSITION_LOG_BY_ADDRESS has established a log position for this instance of open.

User Action: Position log for GETN.

--ERROR AA 2312-- {text} cannot be done because the log record is longer than the user's working_storage_length {text}.

Condition Identifier: AAE\$LOG_RECORD_EXCEEDS_USER_WSL Code: AA 2312

Description: Operation cannot be done, because the log record is longer than the user working_storage_length.

User Action: Correct the log record length.

--ERROR AA 2313-- {text} cannot be done because the log was not opened with READ permission.

Condition Identifier: AAE\$LOG_NOT_OPENED_FOR_READ Code: AA 2313

Description: Operation cannot be done, because the log was not opened with READ permission.

User Action: Log must be opened with READ permission.

--ERROR AA 2314-- TSP\$POSITION_LOG_BY_ADDRESS cannot be done because the specified physical address does not lie within the log.

Condition Identifier: AAE\$INVALID_PHYSICAL_ADDRESS Code: AA 2314

Description: TSP\$POSITION_LOG_BY_ADDRESS cannot be done because the specified address is not in the log.

User Action: Check the specified physical address.

--ERROR AA 2315-- {text} failed because the ring of the log record to be returned differs from the ring of the caller {text}.

Condition Identifier: AAE\$RECORD_RING_TOO_PRIVILEGED Code: AA 2315

Description: Operation failed because the ring of the log record differs from the ring of the caller of the log record that is to be returned.

User Action: Verify ring access to the log file.

--ERROR AA 2316-- time is positioned before BOI.

Condition Identifier: AAE\$TIME_BEFORE_LOG_BOI Code: AA 2316

Description: Time is positioned before beginning-of-information.

User Action: None.

--ERROR AA 2317-- GETN has reached EOI.

Condition Identifier: AAE\$GETN_HAS_REACHED_EOI Code: AA 2317

Description: The GETN command has reached the EOI.

User Action: Reposition the file to the beginning-of-information.

--ERROR AA 2318-- GETP has reached BOI.

Condition Identifier: AAE\$GETP_HAS_REACHED_BOI Code: AA 2318

Description: GETP has reached the beginning-of-information.

User Action: None.

--ERROR AA 2319-- record is not found.

Condition Identifier: AAE\$RECORD_NOT_FOUND Code: AA 2319

Description: The record requested was not found.

User Action: Verify correct record was specified.

--ERROR AA 2320-- The Log_Control_File is invalid; notify the Log Administrator, who must either reconfigure the log or restore it from a backup.

Condition Identifier: AAE\$INVALID_LOG_CONTROL_FILE Code: AA 2320

Description: The LOG_CONTROL_FILE is invalid. Notify the site Log Administrator to reconfigure the log or restore it from backup.

User Action: Notify the Log Administrator for information.

--ERROR AA 2321-- The Log_Journal cannot be accessed; notify the Log Administrator, who probably needs to create the Log_Journal.

Condition Identifier: AAE\$CANT_ACCESS_LOG_JOURNAL Code: AA 2321

Description: The LOG_JOURNAL cannot be accessed. Notify the Log Administrator to create a new LOG_JOURNAL.

User Action: Notify the Log Administrator to create a new log.

--ERROR AA 2322-- Repository {text} cannot be accessed.

Condition Identifier: AAE\$CANT_ACCESS_REPOSITORY Code: AA 2322

Description: The repository cannot be accessed.

User Action: Check that the correct repository was specified.

--ERROR AA 2323-- File {text}: The Log_Control_File is missing; notify the Log Administrator, who must either reconfigure the log or restore it from a backup.

Condition Identifier: AAE\$MISSING_LOG_CONTROL_FILE Code: AA 2323

Description: The log control file is missing. Notify the Log Administrator, who may have to reconfigure the log or restore it from a backup.

User Action: Notify the Log Administrator.

--ERROR AA 2324-- The Tape_Job_Ownership_File cannot be accessed; notify the Log Administrator, who probably needs to create it.

Condition Identifier: AAE\$MISSING_TAPE_JOB_MGMT_FILE Code: AA 2324

Description: The TAPE_JOB_OWNERSHIP_FILE cannot be accessed. Notify the Log Administrator to create it.

User Action: Notify the Log Administrator.

--ERROR AA 2325-- log is already positioned at EOI.

Condition Identifier: AAE\$LOG_POSITIONED_AT_EOI Code: AA 2325

Description: Log is already positioned at EOI.

User Action: None.

--ERROR AA 2326-- Repository {text} was not found on any log backup, so the restoration cannot be done.

Condition Identifier: AAE\$REPOS_NOT_FOUND_ON_BACKUP Code: AA 2326

Description: The repository was not found on any log backup, so the restoration could not be done.

User Action: None.

--ERROR AA 2327-- SCL variable AAV\$LOG_SELECTION must be of type STRING.

Condition Identifier: AAE\$BAD_LOG_SELECTION Code: AA 2327

Description: SCL variable AAV\$LOG_SELECTION must be of type string.

User Action: Correct the SCL variable so that it is of type string.

--ERROR AA 2328-- When REPOSITORIES=VIRTUALS, DATE and TIME must be specified.

Condition Identifier: AAE\$VIRTUAL_REQUIRES_DATE_TIME Code: AA 2328

Description: When REPOSITORIES=VIRTUALS, the date and time must be specified.

User Action: Specify the date and time.

--ERROR AA 2329-- The backing up of a repository failed.

Condition Identifier: AAE\$COPY_TO_BACKUP_FAILED Code: AA 2329

Description: The repository you were trying to backup failed.

User Action: Check that the correct backup was specified. Also, check the error file for possible errors.

--WARNING AA 2330-- A portion of the log ({text}) could not be read.

Condition Identifier: AAE\$GLITCH_IGNORED_IN_LOG Code: AA 2330

Description: A portion of the log could not be read.

User Action: Try reading the log again.

--ERROR AA 2331-- The Backup_Log subcommand of the Administer_Recovery_Log utility was used previously to back up the most recent repository of logging information, and since no further information has been logged since that time, no purpose would be served in backing up the repository currently positioned upon (it is empty). The subcommand is therefore being ignored.

Condition Identifier: AAE\$CANT_BACL_EMPTY_REPOS Code: AA 2331

Description: The currently active repository is empty; it cannot be backed up.

User Action: None.

--ERROR AA 2332-- Log {text}: The log control file for this log cannot be accessed.

Condition Identifier: AAE\$CANT_ACCESS_LOG_CNTL_FILE Code: AA 2332

Description: The indicated log file is currently in use or may not exist.

User Action: If the log is currently in use, try again later. If the log does not exist, use the Restore_Log utility to restore it.

--ERROR AA 2500-- File {file} is not open, so {text} can't be done.

Condition Identifier: AAE\$REQUEST_REQUIRES_OPEN_FILE Code: AA 2500

Description: The file must be open before the specified operation can be performed. This error can occur if a previous open request has failed, which causes all succeeding requests for the file to fail.

User Action: Check that the open request succeeded before attempting to perform the operation.

Further Information: To read about the OPENM call, see the FORTRAN Language Definition Usage manual.

To read about the OPEN statement for keyed files, see the COBOL Usage manual. To read about the FSP\$OPEN_FILE call, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2505-- File{text} {file}{text} : {text} attempted, but the parameter at position {text} is out of range. Check for a syntax error or an invalid data type for the intended FIT field. Also check that character data is delimited correctly.

Condition Identifier: AAES\$VALUE_OUT_OF_RANGE Code: AA 2505

Description: The specified parameter value is outside of the valid range of values.

User Action: Correct the value of the parameter in the given request.

Further Information: Consult the appropriate manual (the COBOL Usage manual, the FORTRAN Language Definition Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual).

--ERROR AA 2510-- File{text} {file}{text} : {text} attempted, but the parameter at position {text} is an unrecognizable keyword.

Condition Identifier: AAES\$UNKNOWN_KEYWORD Code: AA 2510

Description: An unknown keyword is used in the given operation at the specified parameter position.

User Action: Correct the keyword.

Further Information: See the keyed-file interface call description in the FORTRAN Language Definition Usage manual.

--ERROR AA 2515-- File{text} {file}{text} : {text} attempted, but the parameter at position {text} specifies an attribute that can only be fetched, not stored.

Condition Identifier: AAES\$ATTRIBUTE_ONLY_FETCHABLE Code: AA 2515

Description: The specified attribute can only be fetched; it cannot be stored.

User Action: Change the request so that it does not attempt to store a value for the attribute.

Further Information: To learn which attributes can be fetched and which can be stored, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2520-- File{text} {file}{text} : The key position specified by parameters 2 and 3, RKW and RKP, ({text}) is greater than the maximum record length.

Condition Identifier: AAES\$RKW_PLUS_RKP_TOO_BIG Code: AA 2520

Description: As specified by the RKW and RKP values in the FIT, the key position is calculated by multiplying the RKW value by 10 and adding it to the RKP value. As currently specified in the FIT, the RKW and RKP values specify a byte position beyond the end of the record.

User Action: Correct the RKW and/or RKP values.

Further Information: The RKW and RKP values are described in the FORTRAN Language Definition Usage manual.

--ERROR AA 2525-- File{text} {file}{text} : {text} attempted but the parameter in position {text} can't be so used while the file is open.

Condition Identifier: AAES\$ATTRIBUTE_ILLEGAL_WHEN_OPEN Code: AA 2525

Description: The given operation on the specified parameter cannot be done while the file is open. Certain file attributes can be changed only while the file is closed.

User Action: Close the file before attempting the operation. You can change some file attributes with the SCL command CHANGE_FILE_ATTRIBUTES.

Further Information: To read about the CHANGE_FILE_ATTRIBUTES command, see the NOS/VE System Usage manual. To read about file closing and file attribute specification, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual. The NOS/VE Advanced File Management manual also describes keyed-file attributes.

--ERROR AA 2530-- File{text} {file}{text} : {text} attempted, but the record length has not been specified.

Condition Identifier: AA\$NO_RECORD_LENGTH Code: AA 2530

Description: The length of the record is needed for the given operation.

User Action: Specify the number of bytes in the record.

Further Information: See the call or statement description in the COBOL Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2535-- File{text} {file}{text} : {text} attempted to establish {text} in the FIT, but the specified variable is not in COMMON.

Condition Identifier: AA\$KA_OR_WSA_NOT_IN_COMMON Code: AA 2535

Description: The given call detected that the working storage address (WSA), key address (KA), or primary-key address (PKA) as stored in the FIT (or as specified explicitly on the call) is not in a COMMON block.

User Action: FORTRAN optimization (OPTIMIZATION_LEVEL=HIGH) can cause unpredictable results when WSA, KA, and/or PKA are not in a common block. When high optimization is not used, this error may be ignored. Otherwise, if OPTIMIZATION_LEVEL=HIGH is specified on the FORTRAN command, WSA, KA, and PKA should be variables in COMMON.

Further Information: To read about the WSA, KA, and PKA FIT values, see the Keyed-File Interface section of the FORTRAN Usage manual.

--WARNING AA 2540-- File{text} {file}{text} : {text} attempted, but the parameter at position {text} is obsolete or not yet supported. The parameter is ignored and request processing continues.

Condition Identifier: AA\$OUT_OF_DATE_KEYWORD Code: AA 2540

Description: The specified parameter is not valid at this software release.

User Action: Refer to the command or call description to determine the correct parameter names.

Further Information: The valid values are described in the FORTRAN Language Definition Usage manual.

--ERROR AA 2545-- File{text} {file}{text} : cannot open the file because the specified FIT is already used by another instance-of-open.

Condition Identifier: AA\$CANT_OPEN_SAME_FIT_TWICE Code: AA 2545

Description: A FORTRAN program must use a separate file information table (FIT) for each concurrent instance of open of a file.

User Action: Either close the instance of open using the FIT before attempting to open a file using the FIT or create a second FIT with a FILEIS or FILEDA call before the OPENM call.

Further Information: See the Keyed-File Interface section of the FORTRAN Language Definition Usage manual.

--ERROR AA 2550-- File {file}: IFETCH cannot return the requested value due to three possible reasons: the value has not been placed into the FIT by the user (via STOREF or FILEIS), the file is not open, or the value calculated & maintained in the FIT (by the File Interface) is no longer present.

Condition Identifier: AA\$CANT_IFETCH_NON_USER_VALUE Code: AA 2550

Description: The requested FIT value cannot be fetched before the file is opened or after the file is closed unless it has been stored previously by a FILEIS, FILEDA, or STOREF call.

User Action: Check that the correct keyword was specified. If the keyword is correct, change the program, if possible, so that the IFETCH request is issued while the file is open.

Further Information: See the IFETCH description in the FORTRAN Usage manual.

--ERROR AA 2555-- File {file} : {text} encountered more than one key definition with same position and length. Consistent use of one among several duplicate definitions cannot be guaranteed. {text}

Condition Identifier: AA\$FTN_DUPLICATE_KEY_DEF Code: AA 2555

Description: If more than one alternate-key have the same position and length, the intended alternate-key name used in a START, REWIND, or GET call cannot be guaranteed.

User Action: Examine your output carefully for possible alternate-key name discrepancies.

--ERROR AA 2560-- File{text} {file}{text} : {text} attempted but the parameter in position {text} must be in parentheses. One level of parentheses is required for \$OPEN_SHARE_MODES. Two levels are required for \$ACCESS_AND_SHARE_MODES. Check for too many or too few or unbalanced parentheses.

Condition Identifier: AA\$FTN_FULL_PARENTHESES_REQD Code: AA 2560

Description: The format of the \$ACCESS_AND_SHARE_MODES or \$OPEN_SHARE_MODES value specification is incorrect.

User Action: Correct the format of the value specification. The correct format of \$ACCESS_AND_SHARE_MODES is

```
'((access mode),(share mode))[, ..., ((access mode),(share mode))]'
```

The correct format of \$OPEN_SHARE_MODES is

```
'(access mode)[, ..., (access mode)]'
```

Further Information: See the FORTRAN for NOS/VE Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2565-- File{text} {file}{text} : {text} attempted but the parameter in position {text} can't be used. Expecting {text}, found {text}.

Condition Identifier: AA\$FTN_EXPECTING_X_FOUND_Y Code: AA 2565

Description: The format of the keyed-file call is incorrect.

User Action: Check that the keywords and values are in correct format and sequence. The keywords and associated values are:

Keyword	Value
\$ACCESS_AND_SHARE_MODES	'((access mode),(share mode)),..., ((access mode),(share mode))'
\$OPEN_SHARE_MODES	'(access mode), ..., (access mode)'

Further Information: See the FORTRAN for NOS/VE Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2570-- File{text} {file}{text} : {text} attempted but the parameter in position {text} contains the value {text} grouped with another value. This value must always stand alone within the parentheses.

Condition Identifier: AA\$FTN_MUST_STAND_ALONE Code: AA 2570

Description: The format of the \$OPEN_SHARE_MODE or \$ACCESS_AND_SHARE_MODES value specification is incorrect. When the option NONE or ALL is specified, no other options can be specified.

User Action: Correct the access or share modes specification.

Further Information: FORTRAN for NOS/VE Keyed-File and Sort/Merge Interfaces.

--ERROR AA 2600-- File{text} {file}{text} : {text} is not a recognized operation for a file with organization {text}.

Condition Identifier: AA\$UNKNOWN_REQUEST Code: AA 2600

Description: The requested operation is not valid for the file. This error occurs if a file access procedure (FAP) is in use, or an internal interface is called.

User Action: Correct program.

Further Information: Consult the appropriate manual (the COBOL Usage manual, the FORTRAN Language Definition Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual).

--ERROR AA 2605-- File{text} {file}{text} : {text} operation requires a key value, but none was given.

Condition Identifier: AAE\$KEY_REQUIRED Code: AA 2605

Description: Specification of a key value is required to perform the operation.

User Action: Specify a key value.

Further Information: Specification of key values is described in the COBOL Usage manual, the FORTRAN Language Definition Usage manual, and the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2610-- File{text} {file}{text} : {text} found no records in file.

Condition Identifier: AAE\$EMPTY_FILE Code: AA 2610

Description: The file is empty so the requested operation cannot be performed.

User Action: Determine why the file is empty when it was expected to contain data.

Further Information: To read about keyed-file processing, see the COBOL Usage manual, the FORTRAN Language Definition Usage manual, the NOS/VE Advanced File Management Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2615-- File{text} {file}{text} has nonembedded keys, so {text} requires a key value, but none given with this operation.

Condition Identifier: AAE\$NON_EMBEDDED_KEY_NOT_GIVEN Code: AA 2615

Description: Because the primary key is nonembedded, you must specify a primary-key value to perform this operation.

User Action: Specify the primary-key value.

Further Information: Specification of primary-key values is described in the COBOL Usage manual, the FORTRAN Language Definition Usage manual and the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2620-- File{text} {file}{text} : {text} operation needs the location of the WSA (record area), but none was given for this operation.

Condition Identifier: AAE\$WSA_NOT_GIVEN Code: AA 2620

Description: The operation requires a pointer to a working storage area.

User Action: Specify the working storage location on the call, or, for a FORTRAN keyed-file interface call, in the FIT.

Further Information: See the description of the specified call or statement in the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2625-- File{text} {file}{text} : {text} operation not attempted because the WSL (actual record length) of {text} is less than the minimum record length for the file.

Condition Identifier: AAE\$WSL_TOO_SHORT Code: AA 2625

Description: The working storage area must be at least as long as a minimum length record.

User Action: Increase the size of the working storage area or decrease the minimum record length attribute value.

Further Information: See the call description in the FORTRAN Language Definition Usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2630-- File{text} {file}{text} : {text} operation not attempted, because WSL (actual record length) of {text} is above the file maximum record length.

Condition Identifier: AAE\$WSL_TOO_LONG Code: AA 2630

Description: The length of the working storage area (WSL) is greater than the maximum record length.

User Action: Correct the working storage length value or the maximum record length attribute value.

Further Information: See the call description in the FORTRAN Language Definition or CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2635-- File{text} {file}{text} already at {text} position so {text} operation not attempted.

Condition Identifier: AAE\$CANT_POSITION_BEYOND_BOUND Code: AA 2635

Description: The operation attempted to position the file beyond a file boundary (BOI or EOI).

User Action: Correct the program so that it checks for BOI or EOI.

Further Information: To read about positioning a keyed file, see the FORTRAN Usage manual (GETN, SKIP, and REWIND calls), the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2640-- File{text} {file}{text} : {text} operation not attempted, because the major key length of {text} is > the key length for this file.

Condition Identifier: AAE\$MAJOR_KEY_TOO_LONG Code: AA 2640

Description: The specified major-key length is longer than the primary-key length defined for the file. A major key is the first part of the primary key; the system looks for the first primary key that begins with the major key.

User Action: Correct the major-key length so it is less than or equal to the primary-key length. If necessary, enter a DISPLAY_KEYED_FILE_PROPERTIES command to see the primary-key length for the file.

Further Information: To read about using major keys to position a keyed file, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2645-- File{text} {file}{text} : {text} operation tried to fetch an unrecognizable access information item (using the tag in array element : {text}).

Condition Identifier: AAE\$UNKNOWN_ACCESS_INFO_TAG Code: AA 2645

Description: The tag field in one element of the array passed to the operation was not recognized in the context of the operation.

User Action: Correct the tag field value.

Further Information: To see the valid tag field values, refer to the call description in the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2650-- File{text} {file}{text} requires {text} access for the {text} operation.

Condition Identifier: AAE\$NOT_ENOUGH_PERMISSION Code: AA 2650

Description: The file has not been opened with the appropriate access to perform the given operation.

User Action: Specify the indicated access mode in the set of access permissions requested. Access modes can be specified on the open request, on an ATTACH_FILE command, and on a SET_FILE_ATTRIBUTES command.

Further Information: SCL access mode specification: See the SET_FILE_ATTRIBUTES or ATTACH_FILE command descriptions in the NOS/VE System Usage manual.

FORTRAN access mode specification: See the OPENM Call description in the FORTRAN Language Definition Usage manual.

COBOL access mode specification: See the indexed sequential file open description in the COBOL Usage manual.

CYBIL access mode specification: See the access_mode attribute description and the AMP\$FILE and AMP\$OPEN call descriptions in the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2660-- File{text} {file}{text} : operation {text} couldn't be done because SYSTEM_KEY file has reached the block limit imposed by its key size.

Condition Identifier: AAE\$S_K_FILE_AT_BLOCK_LIMIT Code: AA 2660

Description: This operation could not be done because the SYSTEM_KEY file has reached the block limit imposed by its key size.

User Action: Increase the block limit, if possible.

--ERROR AA 2665-- File{text} {file}{text} : {text} can't be done because the preceding file update operation may have changed the DIRECT_ACCESS file structure. Sequential get (GET_NEXT) calls are not allowed after an update (PUT, REPLACE, or DELETE) operation until the file has been repositioned at its beginning by a REWIND or OPEN operation or a SELECT_KEY operation that changes the currently-selected key.

Condition Identifier: AAESCANT_DA_GETN_AFTER_PUT Code: AA 2665

Description: The sequential read of a direct-access file using its primary key cannot continue after an update operation (except a replacement with a record of the same length).

User Action: To begin a sequential read of a direct-access file, position the file at its beginning. The file is initially positioned at its beginning when it is opened and when the primary key is selected after use of an alternate key. Otherwise, a rewind operation is required to position the file at its beginning.

Further Information: See the direct-access file use discussion in the COBOL Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2670-- File{text} {file}{text} : AMP\$DELETE_NESTED_FILE will not work on \$MAIN_FILE, which cannot be deleted.

Condition Identifier: AAESCANT_DELETE_MAIN_NESTED_F Code: AA 2670

Description: The default nested file (\$MAIN_FILE) cannot be deleted from a file. Deleting \$MAIN_FILE is the same as deleting the file itself.

User Action: Correct the nested-file name on the delete request or issue a request to delete the entire file (a PFP\$PURGE call or DELETE_FILE command).

Further Information: See the AMP\$DELETE_NESTED_FILE description in the CYBIL Keyed-File and Sort/Merge Interfaces manual or the PFP\$PURGE description in the CYBIL File Management manual or the DELETE_FILE description in the NOS/VE System Usage manual.

--ERROR AA 2675-- File{text} {file}{text} : {text} can't be done because there is no nested file with the given name, {text}, in the current file.

Condition Identifier: AA\$NESTED_FILE_NOT_FOUND Code: AA 2675

Description: The operation cannot be performed either because the specified nested file does not exist in the file or because a deletion request is pending for the nested file.

User Action: Check that the specified nested-file name is correct. To list the nested files in a file, enter a DISPLAY_KEYED_FILE_PROPERTIES command for the file.

Further Information: See the nested file discussion in the CYBIL Keyed-File and Sort/Merge Interfaces manual or the NOS/VE Advanced File Management Usage manual. The DISPLAY_KEYED_FILE_PROPERTIES command is described in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2680-- File{text} {file}{text} : {text} can't be done because the task has requested at least one alternate key creation or deletion for the current nested file, but has not applied or abandoned the alternate key definition changes. The task must apply or abandon the changes to this nested file before beginning operations on another nested file.

Condition Identifier: AA\$NO_SELECT_DURING_KEYDEF Code: AA 2680

Description: The operation cannot be done because the task has at least one alternate-key definition or deletion pending for the current nested file, but has not yet applied or abandoned the definitions and deletions.

User Action: Either apply or abandon the definitions before the select request.

Further Information: To read about alternate-key definition and selection, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2685-- File{text} {file}{text} : {text} couldn't be completed because the specified working_storage_area is too small to contain all nested-file or alternate-key definitions requested. As many definitions as could fit in the area have been returned. To get all definitions, the caller should try again, specifying a larger working_storage_area. If fetching nested-file definitions, the nested_file_count value returned by the call gives the number of nested files in the file. This value can be used to allocate an area large enough for all nested-file definitions.

Condition Identifier: AAESTOO_LITTLE_SPACE Code: AA 2685

Description: The operation could not be done because the area supplied to receive the information was too small.

User Action: Increase the size of the area to receive the information. If the call returned the number of definitions, use that number to calculate the required space.

Further Information: The KEYLIST and KLCOUNT calls are described in the FORTRAN Language Definition manual. The AMP\$GET_KEY_DEFINITIONS and AMP\$GET_KEY_LIST calls are described in the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2690-- File{text} {file}{text} : {text} can't be done because repeating groups in an alternate key definition are incompatible with FIRST_IN_FIRST_OUT (FIFO) handling of duplicate key values and alternate key concatenation.

Condition Identifier: AAESNO_REPEATING_GROUPS Code: AA 2690

Description: An alternate key cannot be a field in a repeating group and a concatenated key. It also cannot be a field in a repeating group and use first in, first out (FIFO) duplicate key control.

User Action: Correct the alternate key definition so that the repeating groups and concatenated key attributes are not both selected and the repeating groups and FIFO duplicate key control are not both selected.

Further Information: To read about alternate-key definition, see the COBOL Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2695-- File{text} {file}{text} : {text} can't be done because an alternate key named {text} already exists or because an alternate-key creation request is already pending for the same name.

Condition Identifier: AAESCANT_CREATE_EXISTING_NAME Code: AA 2695

Description: Each alternate key defined in a file must have a name different from the names of all other alternate keys in the file and different from all pending alternate-key definitions.

User Action: Either change the name in the alternate-key definition or discard the pending alternate-key definition or request deletion of the existing alternate key with the same name.

Further Information: To read about alternate-key definition, see the COBOL Usage manual (ALTERNATE RECORD KEY clause), the NOS/VE Advanced File Management Usage manual (CREATE_ALTERNATE_INDEXES utility), or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$CREATE_KEY_DEFINITION, AMP\$APPLY_KEY_DEFINITIONS, and AMP\$ABANDON_KEY_DEFINITIONS).

--ERROR AA 2700-- File{text} {file}{text} : {text} can't be done because an alternate key named {text} does not exist or because an alternate-key deletion request is already pending for the same name.

Condition Identifier: AAESCANT_DELETE_MISSING_NAME Code: AA 2700

Description: The operation cannot be performed either because an alternate key with the specified name is not defined in the file or because an alternate-key deletion request is pending.

User Action: Check that the alternate-key name is correct.

Further Information: To read about alternate-key selection, see the COBOL Usage manual (ALTERNATE RECORD KEY Clause), the FORTRAN Language Definition manual (Alternate Key Selection), or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$DELETE_KEY_DEFINITION, AMP\$APPLY_KEY_DEFINITIONS, and AMP\$ABANDON_KEY_DEFINITIONS).

--ERROR AA 2705-- File{text} {file}{text} : {text} did nothing because no alternate-key creation or deletion requests were pending for the file.

Condition Identifier: AAE\$NO_DEFINITIONS Code: AA 2705

Description: No alternate-key definitions or deletions were pending so the request did nothing.

User Action: None required. Specify one or more alternate-key definitions and/or deletions before requesting that the definitions and deletions be applied or abandoned.

Further Information: To read about alternate-key definition or deletion, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2710-- File{text} {file}{text} : {text} can't be done because no alternate key named {text} was ever created for this file.

Condition Identifier: AAE\$ALTKEY_NAME_NOT_FOUND Code: AA 2710

Description: The specified alternate key has not been defined in the file.

User Action: Check that the correct alternate-key name was specified.

Further Information: To read about alternate key selection, see the COBOL Usage manual (ALTERNATE RECORD KEY clause), the FORTRAN Language Definition manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$CREATE_KEY_DEFINITION).

--ERROR AA 2715-- File{text} {file}{text} : {text} can't request automatic unlock with Preserve_Content lock intent.

Condition Identifier: AAE\$NO_AUTO_UNLOCK_PC Code: AA 2715

Description: A Preserve_Content lock (to keep a record from changing before it is read) cannot be requested with the automatic unlock feature. A Preserve_Content lock can be cleared only by an explicit unlock request or by closing the instance-of-open that owns the lock.

User Action: Change the lock request so that either its lock_intent is not Preserve_Content or it does not request automatic unlock. Instead of using automatic unlock, you could add an unlock request to clear the lock after the reason for its existence has passed.

Further Information: See the lock processing discussion in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2720-- File{text} {file}{text} : there is already a nested file called {text} in this file.

Condition Identifier: AAE\$DUP_NESTED_FILE_NAME Code: AA 2720

Description: The nested file cannot be created because another nested file already exists by that name in the file.

User Action: Either change the name of the new nested file or replace the existing nested file by deleting it and creating the new nested file.

Further Information: See the AMP\$CREATE_NESTED_FILE and AMP\$DELETE_NESTED_FILE calls in the CYBIL Keyed-File and Sort/Merge Interfaces manual or the COPY_KEYED_FILE and CREATE_NESTED_FILE command descriptions in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2725-- File{text} {file}{text} : {text} failed because a collate table named {text} can't be found in any object library in the object library list.

Condition Identifier: AAE\$CANT_GET_COLLATE_TABLE Code: AA 2725

Description: The operation failed because the system could not find the specified collation table in the object libraries in the current object library list.

User Action: Check that the collation table name is correct and that the object library containing the collation table is in the object library list. To display the object library list, enter a DISPLAY_PROGRAM_ATTRIBUTES command; to add an object library to the list, enter a SET_PROGRAM_ATTRIBUTES command; to display the contents of an object library, enter a DISPLAY_OBJECT_LIBRARY command.

Further Information: To read about program attribute and object library concepts and commands, see the NOS/VE Object Code Management manual. To read how to create and specify a collation table, see the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2730-- File{text} {file}{text} : {text} failed because {text} is not allowed as a name. A name must begin with a nondigit contain only letters, digits, and the special characters _, @, #, \$, [,], \, ^.

Condition Identifier: AAEBAD_NAME Code: AA 2730

Description: The operation failed because the specified value is not a valid NOS/VE name. A NOS/VE name begins with a letter and contains only letters, digits, and the characters # _ \$ @ [] { } \ | ~ ^.

User Action: Correct the name.

--ERROR AA 2735-- File{text} {file}{text} : A pending deletion request exists for alternate key {text} so you cannot select it by doing {text}. A request to abandon the pending alternate key creation and deletion requests would allow selection of the key.

Condition Identifier: AAENO_SELECT_ON_PENDING_DELETE Code: AA 2735

Description: The attempt to select the alternate key failed because an alternate-key deletion request is pending that would delete the specified alternate key.

User Action: Check that the correct alternate key is specified. If appropriate, abandon the pending alternate-key deletion request and then select the alternate key.

Further Information: To read about alternate-key selection and deletion, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2740-- File{text} {file}{text} : {text} on key {text} is not allowed because that key is currently selected.

Condition Identifier: AAENO_DELETE_CURRENT_KEY Code: AA 2740

Description: The requested operation failed because the alternate key is the currently selected key.

User Action: Select another alternate key or the primary key before requesting the operation.

Further Information: To read about alternate-key deletion and selection, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2745-- File{text} {file}{text} : {text} on nested_file {text} is not allowed because that nested_file is currently selected.

Condition Identifier: AAENO_DELETE_CURRENT_NESTED_F Code: AA 2745

Description: The requested operation failed because the specified nested file is currently selected.

User Action: Select another nested file before issued the request. (Selecting \$MAIN_FILE while deleting is recommended because \$MAIN_FILE cannot be deleted.)

Further Information: To read about nested-file deletion, see the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2750-- File{text} {file}{text} : {text} failed because the ends of the range are the same or the high end is below the low end.

Condition Identifier: AAESHIGH_END_NOT_ABOVE_LOW_END Code: AA 2750

Description: When specifying a range, the first value must be less than the second value. You cannot specify a single value as a range.

User Action: Correct the range.

Further Information: To read about the KLCOUNT and KEYLIST calls, see the FORTRAN Language Definition Usage manual. See the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual for descriptions of AMP\$START AMP\$GET_PRIMARY_KEY_COUNT and AMP\$GET_PRIMARY_KEY_LIST.

--ERROR AA 2755-- File{text} {file}{text} : {text} cannot retrieve a list of primary-key values because the beginning of the range (as defined by FILE_POSITION) does not precede the end of the range (as defined by HIGH_KEY and HIGH_KEY_RELATION). This includes the case where file_position is BOI, high_key is the lowest key in the alternate index, and high_key_relation is EQUAL_KEY or GREATER_OR_EQUAL_KEY.

Condition Identifier: AAE\$HIGH_END_BELOW_CURRENT Code: AA 2755

Description: The file is currently positioned at or after the given high key so the operation cannot move from the current position in the file towards the limit indicated by the high key.

The end case for this is when the file is positioned at its beginning and the specified high key is the lowest key value for the alternate key.

User Action: Either specify a high key value beyond the current position of the file or reposition the file so that the given high key is beyond the file position.

Further Information: To read about the KLCOUNT and KEYLIST calls, see the FORTRAN Language Definition Usage manual. See the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual for descriptions of AMP\$START AMP\$GET_PRIMARY_KEY_COUNT and AMP\$GET_PRIMARY_KEY_LIST.

--ERROR AA 2760-- File{text} {file}{text} : {text} failed because the file must be positioned by an alternate key and the file is now positioned by the primary key.

Condition Identifier: AAE\$NOT_POSITIONED_BY_ALTKEY Code: AA 2760

Description: The operation failed because the primary key is currently selected.

User Action: Select the appropriate alternate key before attempting the operation.

Further Information: To read about alternate key selection, see the COBOL Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2765-- File{text} {file}{text} : The pending alternate key definition and deletion requests must first be applied to the file before {text} will work for key {text}.

Condition Identifier: AAE\$CANT_SELECT_UNTIL_APPLIED Code: AA 2765

Description: The alternate-key definition must be applied to the file before the operation can be performed.

User Action: Apply the alternate-key definition to the file.

Further Information: To read about the application of alternate-key definitions, see the NOS/VE Advanced File Management manual (APPLY_KEY_DEFINITIONS) or the CYBIL Keyed-File and Sort/Merge Interfaces manual (AMP\$APPLY_KEY_DEFINITIONS).

--ERROR AA 2770-- File{text} {file}{text} : {text} failed because KEY_LENGTH + KEY_POSITION of {text} is beyond the MINIMUM_RECORD_LENGTH (or SPARSE_KEY_CONTROL_POSITION is beyond the MINIMUM_RECORD_LENGTH). For repeating group keys with a fixed count, all key values must fit within the MINIMUM_RECORD_LENGTH.

Condition Identifier: AAE\$ALT_KEY_PAST_MINRL Code: AA 2770

Description: The minimum record length is too short to include one or more of the following: all defined key fields, the sparse-key control character, and, if the alternate key is in a repeating group and the key repeats a fixed number of times, all alternate-key values.

User Action: To see the minimum record length (MINRL) attribute value for the file, enter a DISPLAY_KEYED_FILE_PROPERTIES command. Check that the alternate key position and length were specified correctly in the alternate-key definition. If necessary, re-create the file with a longer minimum record length, or choose another field as the alternate key.

Further Information: To see the DISPLAY_KEYED_FILE_PROPERTIES description, see the NOS/VE Advanced File Management Usage manual. To read about alternate-key definition, see the NOS/VE Advanced File Management manual (CREATE_ALTERNATE_INDEXES utility), the COBOL Usage manual (ALTERNATE_RECORD KEY clause) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$APPLY_KEY_DEFINITIONS).

--ERROR AA 2775-- File{text} {file}{text} : {text} failed because the sum of the lengths of the pieces of a concatenated key exceeds 700 or the number of pieces exceeds 64.

Condition Identifier: AA\$CONCATENATED_KEY_TOO_BIG Code: AA 2775

Description: The key definition specifies the record fields to be concatenated as the alternate key. Either the combined length of the fields exceeds 700 bytes or the number of fields exceeds 64.

User Action: Correct the key definition so that the concatenated key does not exceed the limits.

Further Information: To read about concatenated keys, see the NOS/VE Advanced File Management Usage manual (CREATE_KEY_DEFINITION) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2780-- File{text} {file}{text} : a duplicate key value was found while building an index, which had specified NO_DUPLICATES_ALLOWED. To remedy the situation, the faulty index is discarded; it will be built again allowing duplicate values, ORDERED_BY_PRIMARY_KEY, for {text}.

Condition Identifier: AA\$ENABLE_ALTKEY_DUPLICATES Code: AA 2780

Description: The alternate-key definition specified that duplicate key values were not allowed for the alternate key. However, a duplicate key value was found. The error limit has not yet been reached so the alternate-key definition is changed to allow duplicate key values, ordered by primary-key value. The alternate index is rebuilt, using the changed alternate-key definition.

User Action: If an alternate key that allows duplicate key values is not desired, delete the alternate key. This can be done easily using the CREATE_ALTERNATE_INDEXES utility.

Further Information: To read about duplicate key control and deleting alternate keys, see the NOS/VE Advanced File Management manual (CREATE_ALTERNATE_INDEXES) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2785-- File{text} {file}{text} : {text} failed because {text} is currently in use. If this is a delete_nested_file, any position you might have established in it has now been lost.

Condition Identifier: AA\$NO_DELETE_RASP_IN_USE Code: AA 2785

Description: The deletion cannot be performed while the alternate key or nested file is selected by any instance of open.

User Action: Try again later. If appropriate, attach the file for exclusive access (share_modes=none) before attempting the deletion.

Further Information: See the ATTACH_FILE command description in the NOS/VE System Usage manual.

--ERROR AA 2790-- File{text} {file}{text} : {text} operation was frustrated because sequential reading of a direct-access file by primary key requires either exclusive access to the file or a lock on the file. This instance of open used a lock on the file, but the lock expired. The file has been rewound.

Condition Identifier: AA\$DA_GETN_LOST_FILE_LOCK Code: AA 2790

Description: The sequential read of a direct-access file failed because its lock expired and another instance-of-open requested conflicting access to the file.

User Action: Either increase the LOCK_EXPIRATION_TIME attribute value for the file using a CHANGE_FILE_ATTRIBUTES command or attach the file for exclusive access (share_modes=none) so that the file is not shared.

Further Information: For further information on keyed-file sharing, see the COBOL Usage, FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge Interfaces manual. For a description of the CHANGE_FILE_ATTRIBUTES command, see the NOS/VE System Usage or NOS/VE Commands and Functions manual.

--ERROR AA 2795-- File{text} {file}{text} : {text} operation completed but left key {text} locked without a time limit because an expired lock on a different lock frustrated the relock with time limit.

Condition Identifier: AAES\$UNLIMITED_KEY_LOCK_LEFT Code: AA 2795

Description: The granted lock renewal does not have an expiration time because an expired lock exists.

User Action: Clear the expired lock by either issuing an UNLOCK ALL request to clear all locks held by the instance of open or by unlocking each lock one by one until the status aae\$key_last_exp_unlocked (AA2850) is returned. If possible, determine why the lock expired. If appropriate, re-create the file with a larger (or 0 for unlimited) LOCK_EXPIRATION_TIME attribute.

Further Information: See the UNLOCKK call and \$LOCK_EXPIRATION_TIME FIT Value descriptions in the FORTRAN Usage manual or the AMP\$UNLOCK_KEY call and LOCK_EXPIRATION_TIME attribute description in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2800-- File{text} {file}{text} : {text} operation failed because a lock on nested file {text} can't be obtained within the wait time limit. An update may block an apply-key-definitions, or vice versa, and an open may block either of them.

Condition Identifier: AAESCANT_LOCK_NESTED_FILE Code: AA 2800

Description: The operation failed because it could not get a lock on the nested file.

User Action: Try again later. If possible, attach the file for exclusive access (share_modes=none) so that the file is not shared while you use it.

Further Information: For more information on keyed-file sharing, see the description in the COBOL Usage, FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge manual.

--ERROR AA 2805-- File{text} {file}{text} : {text} operation failed because the primary-key value (or the file) couldn't be locked due to an expired key lock from a previous operation.

Condition Identifier: AAES\$KEY_EXPIRED_LOCK_EXISTS Code: AA 2805

Description: The lock request failed because a lock held by the instance of open has expired.

User Action: Clear the expired lock by either issuing an UNLOCK ALL request to clear all locks held by the instance of open or by unlocking each lock one by one until the status aae\$key_last_exp_unlocked (AA2850) is returned. If possible, determine why the lock expired. If appropriate, re-create the file with a larger (or 0 for unlimited) LOCK_EXPIRATION_TIME attribute.

Further Information: See the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2810-- File{text} {file}{text} : {text} operation failed because, though the key was already locked with a time limit, it couldn't be locked with no limit due to an expired key lock from a previous operation.

Condition Identifier: AAES\$EXPIRED_LOCK_INTERFERED_1 Code: AA 2810

Description: The lock request failed because the instance of open holds a lock that has expired.

User Action: Clear the expired lock by either issuing an UNLOCK ALL request to clear all locks held by the instance of open or by unlocking each lock one by one until the status aae\$key_last_exp_unlocked (AA2850) is returned. If possible, determine why the lock expired. If appropriate, re-create the file with a larger (or 0 for unlimited) LOCK_EXPIRATION_TIME attribute.

Further Information: See the UNLOCKK call and \$LOCK_EXPIRATION_TIME FIT value descriptions in the FORTRAN Usage manual or see the AMP\$UNLOCK_KEY call and LOCK_EXPIRATION_TIME attribute descriptions in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2815-- File{text} {file}{text} : {text} operation failed because the key lock table has reached its maximum size. A task must release a lock in this nested file before a new operation can be done.

Condition Identifier: AA\$KEY_LOCK_TABLE_FULL Code: AA 2815

Description: The maximum number of concurrent locks exist for the file.

User Action: Release any locks held for the file and try again.

If possible, change the program so that it releases the locks it holds as soon as possible. If the program requires multiple locks, it should request a file lock.

If possible, attach the file for exclusive access (share_modes=none).

--ERROR AA 2820-- File{text} {file}{text} : {text} failed because there is an expired lock on a key value (or file) left over from an earlier operation.

Condition Identifier: AA\$KEY_EXPIRED_LOCK_EXISTS_2 Code: AA 2820

Description: An expired lock owned by this instance of open prevents further operations for this instance of open until the expired lock is cleared. (The lock is expired because the time limit specified by the LOCK_EXPIRATION_TIME file attribute has passed and another task has been granted a request that would otherwise have been prevented by the expired lock.)

User Action: Clear the expired lock by either issuing an UNLOCK ALL request to clear all locks held by the instance of open or by unlocking each lock one by one until the status aae\$key_last_exp_unlocked (AA2850) is returned. If possible, determine why the lock expired. If appropriate, re-create the file with a larger (or 0 for unlimited) LOCK_EXPIRATION_TIME attribute.

Further Information: See the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2825-- File{text} {file}{text} : {text} by primary key is not allowed on DIRECT_ACCESS files.

Condition Identifier: AA\$NO_SKIP_IN_DA Code: AA 2825

Description: The specified operation is not valid for a direct-access file when its primary key is selected.

User Action: Remove the call or select an alternate key before the call.

Further Information: To read about direct-access file use, see the NOS/VE Advanced File Management Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2830-- File{text} {file}{text} : {text} failed because a lock could not be obtained on alternate key {text}. Instead, the primary key (\$PRIMARY_KEY) has been selected.

Condition Identifier: AA\$CANT_SELECT_KEY Code: AA 2830

Description: The specified alternate key could not be selected due to an internal problem. The primary key was selected instead.

User Action: Try selecting the alternate key again. If the problem re-occurs, notify site personnel. If appropriate, try deleting and re-creating the alternate key.

Further Information: See the description of deleting and creating alternate keys with the CREATE_ALTERNATE_INDEXES utility in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2835-- File{text} {file}{text} : {text} failed because the name {text} has already been reserved by another task as the name of an alternate key to be created in this nested file.

Condition Identifier: AA\$NAME_ALREADY_RESERVED Code: AA 2835

Description: The specified name could not be used because another task has used that name in a pending alternate-key definition for the file.

User Action: Specify another name.

Further Information: To read about alternate-key definition, see the COBOL Usage manual (ALTERNATE RECORD KEY Clause), the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$CREATE_KEY_DEFINITION call), or the NOS/VE Advanced File Management manual (CREATE_ALTERNATE_INDEXES utility).

--ERROR AA 2840-- File{text} {file}{text} : {text} failed because some other task has the list of rasps (a system internal table) locked.

Condition Identifier: AAESCANT_LOCK_LIST_OF_RASPS Code: AA 2840

Description: The operation failed due to an internal error.

User Action: Try again later. If the problem re-occurs, report it to site personnel.

--ERROR AA 2845-- File{text} {file}{text} : {text} failed because nested file {text} is locked for Exclusive_Access by another task. Instead, the default nested file (\$MAIN_FILE) has been selected.

Condition Identifier: AAESCANT_SELECT_NESTED_FILE Code: AA 2845

Description: The nested file could not be selected because another task has an Exclusive_Access file lock on the nested file.

User Action: Try again later. If appropriate, attach the entire file for exclusive access (share_modes=none).

Further Information: See the ATTACH_FILE command description in the NOS/VE System Usage manual. File locks are described in the FORTRAN Usage and CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2850-- File{text} {file}{text} : {text} succeeded on key {text}, whose lock had expired. At this time, no expired locks remained.

Condition Identifier: AAESKEY_LAST_EXP_UNLOCKED Code: AA 2850

Description: All expired locks have been cleared. File operations can resume for this instance of open.

User Action: None required. When locks frequently expire, you should check the LOCK_EXPIRATION_TIME attribute for the file. If appropriate, re-create the file with a larger (or 0 for unlimited) lock expiration time attribute.

--ERROR AA 2852-- {text} requires {file} to be an existing file.

Condition Identifier: AAESEXISTING_FILE_REQUIRED Code: AA 2852

Description: You attempted to display a lock file connection to a keyed file that does not exist.

User Action: Specify an existing keyed file.

--ERROR AA 2853-- {text} requires you to have {text} access to file {text}.

Condition Identifier: AAESUSER_NOT_PERMITTED_TO_FILE Code: AA 2853

Description: To establish a lock file connection or clear a lock file connection, you must have control permission for both the keyed file and the lock file.

User Action: Establish control permissions for the keyed file and the lock file using the CHANGE_FILE_PERMITS command.

--ERROR AA 2855-- File{text} {file}{text} : {text} succeeds on key {text}, whose lock had expired. At least one more expired lock remains.

Condition Identifier: AAESKEY_NON_LAST_EXP_UNLOCKED Code: AA 2855

Description: The expired lock specified on the request has been cleared, but the instance of open owns at least one more expired lock.

User Action: Continue clearing locks individually or issue an UNLOCK ALL request. The expired lock may be for the currently selected nested file or for another nested file in the file.

Further Information: To read about the UNLOCK 'ALL' call, see the FORTRAN Usage manual. To read about the AMP\$UNLOCK_KEY ALL call, see the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2856-- Lock file {text} cannot be a private lock file.

Condition Identifier: AAESLOCK_FILE_CANNOT_BE_PRIVATE Code: AA 2856

Description: You tried to specify the default lock file, \$SYSTEM.AAM.AAF\$LOCK_FILE, as a private lock file.

User Action: Either use the default lock file as a public lock file or create a private lock file and connect your keyed file to the new private lock file.

--ERROR AA 2857-- Lock file {text} is not connected to keyed file {text}.

Condition Identifier: AA\$KEYED_FILE_NOT_PAIRED_TO_LF Code: AA 2857

Description: Either the current lock file is not connected to any keyed file, or the keyed file you specified is connected to a different lock file.

User Action: Either establish a connection between the specified lock file and keyed file, or clear the connection between the keyed file and the other lock file and establish a connection to the new lock file.

--ERROR AA 2858-- {text} cannot clear the connection between lock file {text} and the keyed file because the lock file is connected to {text}.

Condition Identifier: AA\$PRIVATE_LOCK_FILE_MISMATCH Code: AA 2858

Description: You used CLEAR_LOCK_FILE_CONNECTION and the current lock file is not connected to the specified keyed file.

User Action: Determine which lock file is connected to the specified keyed file with the DISPLAY_LOCK_FILE_CONNECTION subcommand. You can also determine which keyed file or files is connected to the current lock file with the DISPLAY_LOCK_FILE subcommand.

--ERROR AA 2859-- {text} cannot {text} {text} because {text}.

Condition Identifier: AA\$CANT_PROCESS_CONN_REQUEST Code: AA 2859

Description: You cannot display a lock file which does not exist. You cannot establish a lock file connection between a keyed file and a lock file if the lock file does not exist or if the lock file is private and the keyed file is public.

User Action: Create a lock file or make the existing lock file public

--ERROR AA 2860-- File{text} {file}{text} : {text} failed because the KEY_TYPE was COLLATED_KEY, but no COLLATE_TABLE_NAME was passed. (No default exists because the file is not an indexed-sequential file with a collated primary key).

Condition Identifier: AA\$COLLATED_ALTKEY_NO_TABLE Code: AA 2860

Description: The alternate key could not be defined as a collated key because no collation table was specified for the alternate key and no default collation table is available (because the file is not an indexed-sequential file with a collated primary key).

User Action: Specify the name of the collation table as an attribute in the alternate-key definition or change its key type to uncollated.

Further Information: To read about alternate key collation table specification, see the NOS/VE Advanced File Management manual (CREATE_KEY_DEFINITION) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$CREATE_KEY_DEFINITION).

--ERROR AA 2861-- {text} cannot {text} {text} because {text}. Release the old connection between the keyed file and the lock file with the CLEAR_LOCK_FILE_CONNECTION subcommand.

Condition Identifier: AA\$CANT_CHANGE_PRIVATE_LF_CONN Code: AA 2861

Description: You tried to establish a connection with a keyed file that is already connected to a private lock file, or you tried to establish a connection to a private lock file that is already connected to a different keyed file.

User Action: Determine which file is already connected with DISPLAY_LOCK_FILE and DISPLAY_LOCK_FILE_CONNECTION. Then, either connect the keyed file to a different lock file or connect the lock file to a different keyed file.

--ERROR AA 2862-- File {text} : {text} requires the file specified, for the {text} parameter, be a keyed file.

Condition Identifier: AA\$KEYED_FILE_REQUIRED Code: AA 2862

Description: You used ESTABLISH_LOCK_FILE_CONNECTION, CLEAR_LOCK_FILE_CONNECTION, or DISPLAY_LOCK_FILE_CONNECTION and you did not specify a keyed file on the subcommand.

User Action: Retry the subcommand, and specify a keyed file on the subcommand.

--ERROR AA 2863-- {text} cannot create lock file {text} because {text}.

Condition Identifier: AAE\$CANT_CREATE_LOCK_FILE Code: AA 2863

Description: You cannot create the specified lock file because a lock file already exists with that name.

User Action: Specify a different name for the lock file you want to create.

--ERROR AA 2864-- {text} may only be used after a lock file has been selected with the Use_Lock_File subcommand.

Condition Identifier: AAE\$LOCK_FILE_NOT_SPECIFIED Code: AA 2864

Description: You attempted to use an Administer_Lock_File subcommand without first specifying a lock file with USE_LOCK_FILE. The only subcommands you can use without first specifying a lock file are HELP, QUIT, and DISPLAY_LOCK_FILE_CONNECTION.

User Action: Specify a lock file with the USE_LOCK_FILE subcommand.

--ERROR AA 2865-- File{text} {file}{text} : {text} found a sparse_key_control character which indicated key inclusion, but the record was too short to contain the key value. The operation continues without a key value for this record.

Condition Identifier: AAE\$SPARSE_KEY_BEYOND_EOR Code: AA 2865

Description: Although the character in the sparse key control position indicates that the alternate-key value for the record should be included in the alternate index, it is not included because the alternate key field extends beyond the end of the record.

User Action: None required. However, you should determine why the record did not contain a complete alternate-key value.

Further Information: To read about sparse-key control and apply error processing, see the NOS/VE Advanced File Management manual or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2866-- File {text} : {text} determined that this is not a usable lock file because {text}.

Condition Identifier: AAE\$LOCK_FILE_INVALID Code: AA 2866

Description:

The lock file you specified is not a usable lock file for one of the following reasons:

- The file does not have ring attributes of (4, 4, 4).
- The file does not contain a correct lock file header.
- The file is not permanent.
- The file does not have a sequential file organization.
- The file is named AAF\$DEPENDENCY_FILE.

User Action: Change the lock file so it is a usable lock file, or, clear any connections, delete the lock file, and create a new one.

--ERROR AA 2867-- File{text} {file}{text} : {text} found a conflicting expired lock in a parcel belonging to another task. It would have expired the parcel, but could not get space allocated for a copy of the parcel.

Condition Identifier: AAE\$NO_SPACE_FOR_STOLEN_PARCEL Code: AA 2867

Description: This is a system internal error.

User Action: Report this to your site analyst.

--ERROR AA 2870-- File{text} {file}{text} : {text} encountered a duplicate key and found that the nonfatal-error limit had been reached. It then discarded any new alternate indexes that it had built (although it cannot restore any alternate indexes it deleted). Had the ERROR_LIMIT not been reached, the alternate-key definition would have been modified to allow duplicates. The duplicate key values relate to {text}.

Condition Identifier: AAE\$DUPLICATE_KEY_LIMIT Code: AA 2870

Description: The limit on duplicate alternate-key values was reached. The newly-created alternate indexes are discarded, but the deleted alternate indexes remain deleted.

User Action: Determine why the data contains more duplicate key values than expected. Check that the alternate-key definitions are correct. If appropriate, define the alternate key so that it allows duplicates.

Further Information: To read about duplicate key control, see the COBOL Usage manual (DUPLICATES phrase), the NOS/VE Advanced File Management Usage manual (APPLY_KEY_DEFINITIONS), or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$APPLY_KEY_DEFINITIONS).

--ERROR AA 2871-- File{text} {file}{text} : {text} failed because this parcel can be committed only with a SIMPLE_COMMIT parameter.

Condition Identifier: AAE\$SIMPLE_COMMIT_NEEDED Code: AA 2871

Description: Currently, the only option to commit a parcel in a CYBIL program is SIMPLE_COMMIT.

User Action: Specify SIMPLE_COMMIT on the AMP\$COMMIT_PARCEL CYBIL interface procedure.

Further Information: See the CYBIL for NOS/VE Keyed-Files and Sort/Merge Interfaces usage manual.

--ERROR AA 2873-- Log {text} is damaged and cannot be used for recovery. Please notify the log administrator that Restore_Log should be performed.

Condition Identifier: AAE\$LOG_RECOVERY_REQUIRED Code: AA 2873

Description: An attempt to read from or write to the log has occurred, however, the log is damaged.

User Action: It is recommended that the log administrator restore the log using the Restore_Log utility.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2874-- Log file {text} could not be restored. The log cannot be enabled.

Condition Identifier: AAE\$CANT_RESTORE_MINOR_FILE Code: AA 2874

Description: An attempt was made to restore the minor log files (the tape ownership file or the log journal), but failed.

User Action: The log is not usable. You must use a different log, or you must delete the log and then recreate it.

--ERROR AA 2875-- File{text} {file}{text} : AMP\$APPLY_KEY_DEFINITIONS {text}.

Condition Identifier: AAE\$ERRORS_DURING_APPLY Code: AA 2875

Description: The indicated error was found while attempting to build an alternate index in the file.

User Action: If the error prevents application of the alternate-key definition, correct the error in the data or the definition and try again.

Further Information: Alternate key creation is described in the CYBIL Keyed-File & Sort/Merge Interfaces manual, the FORTRAN Usage manual, the COBOL Usage manual, and the NOS/VE Advanced File Management manual.

--ERROR AA 2876-- File {text}: When the file to be recovered does not exist RECOVER_KEYED_FILE requires that the log residence be explicitly specified using SET_FILE_ATTRIBUTES.

Condition Identifier: AAE\$LOG_RESIDENCE_REQUIRED Code: AA 2876

Description: If the file to be recovered does not currently exist, you must enter a SET_FILE_ATTRIBUTES command for the file to specify the LOG_RESIDENCE.

User Action: Enter the SET_FILE_ATTRIBUTES command. The command must specify the same file cycle specified on the RECOVER_KEYED_FILE command and the same LOG_RESIDENCE as that of the keyed file before it was updated.

--ERROR AA 2877-- The {text} subcommand was entered but no backups were configured for the log. Please configure a backup for each log repository, and then enter QUIT again.

Condition Identifier: AAE\$SETLBA_REQUIRES_BACKUPS Code: AA 2877

Description: If the SET_LOG_BACKUP_ACCOUNT subcommand is entered during a Administer_Recovery_Log utility session, you must also specify a backup file for each repository with the CONFIGURE_LOG_BACKUP subcommand. The CONFIGURE_LOG_BACKUP subcommand must be entered in the same session in which the SET_LOG_BACKUP_ACCOUNT subcommand was used.

User Action: Enter the CONFIGURE_LOG_BACKUP subcommand for each repository.

--ERROR AA 2878-- There are no alternate key definitions pending to display.

Condition Identifier: AA\$NO_PENDING_DEFS_TO_DISPLAY Code: AA 2878

Description: There are no pending alternate-key creations or deletions to display.

User Action: None.

--ERROR AA 2879-- The log control file did not pass validation. It should be reloaded from a log backup using the Restore_Log_Control_File subcommand. Unfortunately, if there are any entries in the currently active repository, they will be lost when this command completes.

Condition Identifier: AA\$RELOAD_OF_LCF_RECOMMENDED Code: AA 2879

Description: The log control file is damaged and must be restored from the backup file.

User Action: Use the RESTORE_LOG_CONTROL_FILE subcommand of the Restore_Log utility to restore the log control file. Then use the RESTORE_REPOSITORIES subcommand of the Restore_Log utility to restore the damaged repositories. Once the log is restored, it is, however, available only for recovering keyed files. To begin recording updates again, you must switch to a different log, or you must delete the log, then recreate it.

--ERROR AA 2880-- File {text} {file}{text} : {text} fails because the file is DIRECT_ACCESS, its primary key is selected, and the caller does not have either exclusive file access (no share modes) or any lock on the file. A request to lock the file (AMP\$LOCK_FILE) is suggested.

Condition Identifier: AA\$CANT_DA_GETN_IF_SHARED Code: AA 2880

Description: A direct-access file cannot be read sequentially while it could be shared. This is because another instance of open could update the file during the sequential read; the updating could change the file structure, causing the sequential read to miss records or read records more than once.

User Action: Before beginning a sequential read of a direct-access file, attach the file for exclusive access (share_modes=none) or request an exclusive_access lock on the file.

Further Information: For the ATTACH_FILE command description, see the NOS/VE System Usage or NOS/VE Commands and Functions manual. For a description of file locks and the lock request calls, see the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2881-- {text} cannot delete {text} because the damaged flag is not set in the log.

Condition Identifier: AA\$CANT_DELETE_UNDAMAGED_LOG Code: AA 2881

Description: You cannot delete a log control file or a repository unless the log is disabled.

User Action: You can enter the VALIDATE_LOG or the RESTORE_REPOSITORIES subcommands.

If you enter VALIDATE_LOG, and if the log control file or the repository is damaged, the log is immediately disabled. You can then delete the log control file or repository.

If you enter RESTORE_REPOSITORIES, the log is immediately disabled whether the log is damaged or not. You can then delete the log control file or repository.

--ERROR AA 2882-- File {text} could not be recovered because the most recent valid backup found is older than the oldest entry in the user log. This backup was performed on {text}, and is the version of the file now disk resident. To use this version of the file use the VOID_LOG_FOR_RESTORED_FILE subcommand of RECOVER_KEYED_FILE and then backup the file with BACKUP_PERMANENT_FILE to restart logging.

Condition Identifier: AA\$BACKUP_OLDER_THAN_USER_LOG Code: AA 2882

Description: The keyed file that was restored does not correspond to the updates currently in the user log. This is because the restored keyed file is older than the log. A later version could not be restored.

User Action: To begin recording updates on the log for the restored keyed file, you must discard all past logged updates for the keyed file by using the VOID_LOG_FOR_RESTORED_FILE subcommand of the Recover_Keyed_file utility. Then backup the keyed file using the Backup_Permanent_File utility.

--ERROR AA 2883-- File {file} should be deleted by using the {text} subcommand. When this is followed by the {text} subcommand it will cause the file to be restored from the log backup.

Condition Identifier: AA\$PURGE_OF_FILE_RECOMMENDED Code: AA 2883

Description: Based on the status returned by the open, it may be necessary to delete the repository using the DELETE_REPOSITORIES subcommand of the Restore_Log utility. This will force the repository to be restored from the backup files when the RESTORE_REPOSITORIES subcommand is entered.

User Action: Enter the DELETE_REPOSITORIES subcommand to delete the repository, then reenter the RESTORE_REPOSITORIES subcommand.

--ERROR AA 2884-- Repository {file}: This repository could not be restored because {text}.

Condition Identifier: AA\$COULDNT_RESTORE_REPOSITORY Code: AA 2884

Description: The repository you are attempting to restore is either damaged, or was not backed up by the CONFIGURE_LOG_BACKUP subcommand of the Administer_Recovery_Log utility.

User Action: None. The indicated repository cannot be restored.

--ERROR AA 2885-- File{text} {file}{text} : {text} does not allow specification of {text}.

Condition Identifier: AA\$PARM_ILLEGAL_IN_ENVIRONMENT Code: AA 2885

Description: The indicated parameter specification is not valid at this level.

User Action: Correct the parameter specification.

Further Information: See the command description in the SCL Advanced File Management Usage manual or in the SCL Quick Reference manual.

--ERROR AA 2886-- {text} was unable to {text} because of the following error:

Condition Identifier: AA\$BAD_STATUS_FROM_CALL Code: AA 2886

Description: The RESTORE_REPOSITORIES or RESTORE_LOG_CONTROL_FILE subcommands attempted to perform the indicated operation, but failed.

User Action: The message displayed describes the nature of the error.

--ERROR AA 2887-- The ENABLE_LOG subcommand may only be entered after RESTORE_REPOSITORIES is successful.

Condition Identifier: AA\$ENABLE_LOG_NOT_PERMITTED Code: AA 2887

Description: You can enable a log only when the log is ready to be enabled.

User Action: Once the damaged repositories are valid, use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log.

--ERROR AA 2888-- Because {text}, the log control file is not ready for the DAMAGED condition to be reset and the log family is not ready for use in recovery. Enter the RESTORE_REPOSITORIES subcommand.

Condition Identifier: AA\$LOG_NOT_READY_TO_BE_ENABLED Code: AA 2888

Description: You can enable a log only when the RESTORE_REPOSITORIES subcommand has completed successfully.

User Action: Use the RESTORE_REPOSITORIES subcommand of the Restore_Log utility to replace the damaged repositories. Once the damaged repositories are successfully replaced, use the ENABLE_LOG subcommand of the Restore_Log utility to enable the log.

--ERROR AA 2890-- {text} requires the parameter specification APPLY_KEY_DEFINITIONS=YES or APPLY_KEY_DEFINITIONS=NO when alternate key definitions or deletions are pending.

Condition Identifier: AA\$NEED_PARM_WHEN_DEFS_ACTIVE Code: AA 2890

Description: One or more alternate-key definition or deletion requests are still pending so you must specify whether the requests should be applied or canceled before exiting the utility.

User Action: Specify the APPLY_KEY_DEFINITIONS parameter on the QUIT subcommand. To request application, specify APPLY_KEY_DEFINITIONS=YES; to request cancellation, specify APPLY_KEY_DEFINITIONS=NO.

--ERROR AA 2891-- File {text} on log backup {file} is not valid.

Condition Identifier: AA\$BACKUP_OF_LOG_UNUSABLE Code: AA 2891

Description: The log backup is not valid and cannot be used to restore the log control file or the repository.

User Action: If the log control file is damaged, you can use the RESTORE_LOG_CONTROL_FILE subcommand of the Restore_Log utility and specify a different backup. Unfortunately, if a repository is damaged, you cannot restore it.

--ERROR AA 2892-- File {file}: This file could not be opened for segment access.

Condition Identifier: AA\$CANT_GET_SEGMENT_ACCESS Code: AA 2892

Description: The indicated log file cannot be read.

User Action: If this error is displayed while using the RESTORE_REPOSITORIES subcommand of the Restore_Log utility and:

- the indicated log file is a log control file, use RESTORE_LOG_CONTROL_FILE to replace the log control file, or
- the indicated log file is a repository file, delete the repository, then reenter the RESTORE_REPOSITORIES subcommand.

If this error is displayed while using the RESTORE_LOG_CONTROL_FILE subcommand of the Restore_Log utility, delete the log control file, then reenter the RESTORE_LOG_CONTROL_FILE subcommand.

--ERROR AA 2893-- File {file}: This log backup could not be accessed because of the following bad status.

Condition Identifier: AA\$CANT_OPEN_LOG_BACKUP Code: AA 2893

Description: The indicated log backup cannot be opened.

User Action: The message displayed indicates why the log backup cannot be opened.

--ERROR AA 2894-- File {file}: The attempt to read the log backup file returned the following bad status.

Condition Identifier: AA\$CANT_READ_LOG_BACKUP Code: AA 2894

Description: The log backup cannot be read.

User Action: The message displayed indicates why the log backup cannot be read.

--ERROR AA 2895-- File {text} {file}{text} : {text} requires that SPARSE_KEY_CONTROL_CHARACTERS be specified when SPARSE_KEY_CONTROL_POSITION is specified.

Condition Identifier: AA\$MISSING_SPARSE_CHARS Code: AA 2895

Description: When you specify a sparse-key control position, you must also specify a set of sparse-key control characters. The character in the sparse-key control position is compared against the set sparse-key control characters.

User Action: Specify the set of sparse-key control characters.

--ERROR AA 2896-- File {file}: This log backup is not valid because {text}.

Condition Identifier: AAE\$BACKUP_OF_LOG_INVALID Code: AA 2896

Description: The indicated log backup is invalid because:

- the logo of the backup file is not correct. (A logo identifies the file as a log backup file.), or
- the log backup file is not long enough to contain a copy of the log control file, or
- the backup file ends before the end of the repository.

User Action: If the log backup is a log control file, enter the RESTORE_LOG_CONTROL_FILE subcommand of the Restore_Log utility specifying a different backup. Unfortunately, if the log backup is a repository, it cannot be restored.

--ERROR AA 2897-- The {text} parameter may not be specified when {text}.

Condition Identifier: AAE\$ILLEGAL_PARM_SPECIFIED Code: AA 2897

Description: If you are restoring the log control file from disk (MEDIA=MASS_STORAGE_DEVICE), you cannot specify the magnetic tape parameters, RECORDED_VSN, EXTERNAL_VSN, or TYPE, on the RESTORE_LOG_CONTROL_FILE subcommand of the Restore_Log utility.

User Action: Do not specify RECORDED_VSN, EXTERNAL_VSN, or TYPE on the RESTORE_LOG_CONTROL_FILE subcommand.

--ERROR AA 2898-- Log {text}: This is an existing log but it is missing a required repository: {text}. The log is unusable as it currently exists. Either delete and re-create the log or (if this log was configured with log backups) exit this utility and execute the RESTORE_LOG command.

Condition Identifier: AAE\$LOG_REPOSITORY_MISSING Code: AA 2898

Description: The indicated log is missing a required repository; the log cannot be used for recovery purposes or for recording further updates.

User Action: If backups are configured for the log file, use the RESTORE_REPOSITORIES subcommand of the Restore_Log utility to restore the missing repository. Otherwise, you must delete the log, then recreate it.

--ERROR AA 2899-- The file was not restored because there was no tape drive available to mount the backup tape. Either release a tape that is currently allocated, or enter RESERVE_RESOURCE to increase the number of tape drives available to the job.

Condition Identifier: AAE\$NO_TAPE_DRIVE_AVAILABLE Code: AA 2899

Description: The file was not restored because this job requires the use of a tape drive, however, you have a tape drive already in use.

User Action: Either release the tape that is currently allocated to a tape drive, or use the RESERVE_RESOURCE command to specify an additional tape drive. Then execute the utility again. Once the resources (tape drives) are no longer needed, you should release them by entering the RELEASE_RESOURCES command.

--ERROR AA 2900-- File{text} {file}{text} : {text} did not find {text} in the list of pending alternate key definitions or deletions.

Condition Identifier: AAE\$NAME_NOT_IN_KEYDEF_LIST Code: AA 2900

Description: The operation could not be performed because an alternate-key definition is not pending for the specified name.

User Action: Check that the correct name was specified. Check also that the alternate-key definition has not already been applied to the file. (An existing alternate key must be deleted instead of canceled.)

Further Information: For a description of alternate-key definition cancellation, see the NOS/VE Advanced File Management Usage manual (CANCEL_KEY_DEFINITION) or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual (AMP\$CANCEL_KEY_DEFINITION).

--ERROR AA 2901-- The file was not restored because Request_Magnetic_Tape returned the following error:

Condition Identifier: AAESTAPE_ERROR_ON_RESTORE Code: AA 2901

Description: REQUEST_MAGNETIC_TAPE attempted to read the tape on which the backup file resides, but failed.

User Action: The message displayed describes the nature of the failure.

--ERROR AA 2902-- File {text} : {text} cannot add this tape to the backup list because the {text} ({text}) is already in the backup list.

Condition Identifier: AAESVSN_ALREADY_EXISTS Code: AA 2902

Description: The tape specified by the EVSN or RVSN parameter on CONFIGURE_LOG_BACKUP is already being used.

User Action: Specify another tape.

--ERROR AA 2903-- Log {text} : {text} cannot delete the log specified because {text}.

Condition Identifier: AAESCANT_DELETE_LOG Code: AA 2903

Description: The log catalog specified cannot be deleted because it does not exist or is already attached.

User Action: If the log catalog is already attached, try deleting the log catalog again later.

--ERROR AA 2904-- {text} required space on the system heap in ring 4 for {text}, but none was available.

Condition Identifier: AAESCMD_SYSTEM_HEAP_R4_FULL Code: AA 2904

Description: Internal error; the system was unable to allocate space on the system heap for a ring 4 routine.

User Action: Report the software error.

--ERROR AA 2905-- File {file} : {text} detected abnormal status on a call to {text}

Condition Identifier: AAESBAD_STATUS_RETURNED Code: AA 2905

Description: Internal error; a call to an internal routine returned abnormal status.

User Action: Report the software error.

--ERROR AA 2906-- File {text} : Recover_Keyed_File can not recover local files.

Condition Identifier: AAESCANT_RECOVER_LOCAL_FILE Code: AA 2906

Description: The RECOVER_KEYED_FILE command cannot recover local files.

User Action: Make the file a \$USER file.

--ERROR AA 2907-- File {text} : Recover_Keyed_File encountered an internal error: {text}. Recovery was aborted.

Condition Identifier: AAESCMD_ERROR_DURING_RECOVERY Code: AA 2907

Description: The RECOVER_KEYED_FILE Utility encountered an internal error. Recovery was aborted.

User Action: Check with your site administrator.

--ERROR AA 2908-- File {text} : file has been ruined as requested.

Condition Identifier: AAESFILE_RUINED_AS_REQUESTED Code: AA 2908

Description: File has been ruined as requested.

User Action: None.

--ERROR AA 2909-- File {text} : Recover_Keyed_File encountered a system error: {text}.

Condition Identifier: AAES\$SYSTEM_ERROR_IN_RECOVERY Code: AA 2909

Description: The RECOVER_KEYED_FILE Utility encountered a system error.

User Action: Notify the Log Administrator.

--ERROR AA 2910-- File {file} : {text} did not find {text} in the file or in the pending list.

Condition Identifier: AAE\$NAME_NOT_PENDING_OR_APPLIED Code: AA 2910

Description: No alternate-key definition, pending or applied, has the specified name.

User Action: Check that the correct name was specified. Check if the specified alternate-key definition has already been canceled or deleted.

Further Information: For more information on alternate-key creation and deletion, see the NOS/VE Advanced File Management Usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--ERROR AA 2911-- File {text} could not be recovered because the most recent available backup of the file does not have logging enabled.

Condition Identifier: AAE\$RECOVERY_OFF_IN_BACKUP Code: AA 2911

Description: The file could not be recovered because the most recent available backup of the file does not have logging enabled.

User Action: Correct the logging enabled parameter.

--ERROR AA 2912-- File {text} could not be recovered because the log does not contain a log record for a valid backup of this file.

Condition Identifier: AAE\$BACKUP_LOG_RECORD_NOT_FOUND Code: AA 2912

Description: File could not be recovered. Log file does not contain a log record of a valid backup for this file.

User Action: None available.

--ERROR AA 2913-- RECOVER_KEYED_FILE can not recover file {text} because its file organization is not indexed sequential, direct_access or system_key.

Condition Identifier: AAE\$CANT_RECOVER_NON_AAM_FILE Code: AA 2913

Description: The RECOVER_KEYED_FILE Utility cannot recover the specified file, because the file organization is not indexed_sequential, direct_access or system_key.

User Action: File organization must be specified as SEQUENTIAL.

--ERROR AA 2914-- File {text}: When the file to be recovered does not exist RECOVER_KEYED_FILE requires that the file cycle number be explicitly specified as part of the file path.

Condition Identifier: AAE\$FILE_CYCLE_REQUIRED Code: AA 2914

Description: The RECOVER_KEYED_FILE Utility requires that the file cycle be explicitly specified as part of the file path.

User Action: Specify the file cycle as required.

--ERROR AA 2915-- File{text} {file}{text} : {text} failed because of a block lock conflict.

Condition Identifier: AAE\$FLUSH_FAILED_ON_BLOCK_LOCK Code: AA 2915

Description: The attempt to flush the updated file to mass storage failed due to an internal lock on a file block.

User Action: Try again later. If the problem re-occurs, report it to site personnel.

--ERROR AA 2916-- Log {text} : {text} cannot be performed on a log that is not configured for log backups.

Condition Identifier: AAE\$BACKUP_LOG_REQUIRES_CONLB Code: AA 2916

Description: An attempt to backup the log was made. However, the log is not configured for log backups. You cannot backup the log.

User Action: If you want to backup a log, you must configure the log for backups using the CONFIGURE_LOG_BACKUP subcommand in the same session in which the log is created.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2917-- {text} requires {file} to be a sequential file.

Condition Identifier: AA\$FILE_IS_NOT_SEQUENTIAL Code: AA 2917

Description: The specified file must be sequential, but is not.

User Action: If the specified file does not already exist, change the file attributes. If the specified file exists, specify another file.

--ERROR AA 2918-- The validation and initialization of the log backup file failed with the following error:

Condition Identifier: AA\$LOG_BKUP_VALIDATION_ERROR Code: AA 2918

Description: The attempt to create or initialize the log backup file failed.

User Action: Remove the file from the pool of backup files. Specify the REMOVE_FILE parameter on the CONFIGURE_LOG_BACKUP subcommand. Try again, specifying another file on the ADD_FILE parameter of the CONFIGURE_LOG_BACKUP subcommand of the Administer_Recovery_Log utility.

--ERROR AA 2919-- {text} may only be used after the Delete_Log subcommand, or after a catalog has been established with the Use_Log subcommand.

Condition Identifier: AA\$CMD_WITH_NO_CURRENT_LOG Code: AA 2919

Description: The CANCEL_LOG_CHANGES subcommand of the Administer_Recovery_Log utility discards log attributes. It must be preceded by the DELETE_LOG subcommand or the USE_LOG subcommand.

User Action: None. Informative.

--ERROR AA 2920-- File{text} {file}{text} : RMKDEF cannot create an alternate_ key whose position and length match that of the primary_key.

Condition Identifier: AA\$RMKDEF_POS_LEN_MATCHES_PK Code: AA 2920

Description: An RMKDEF call cannot create an alternate key whose position and length are the same as the primary-key position and length. Creation of such an alternate key is disallowed for compatibility with the CYBER 170 RMKDEF call.

User Action: Check that the correct position and length were specified. If appropriate, you can create an alternate key whose position and length match the primary-key position and length using the CREATE_ALTERNATE_INDEXES utility.

Further Information: Use of the CREATE_ALTERNATE_INDEXES utility is described in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2921-- {text}: {text} encountered {text} at PVA {text}.

Condition Identifier: AA\$PVA_OF_UTILITY_SYSTEM_ERROR Code: AA 2921

Description: This is an internal error.

User Action: Report this to your system analyst.

--ERROR AA 2922-- File {file} : {text} cannot create the file specified by the {text} parameter because the file already exists and contains data.

Condition Identifier: AA\$CANT_CREATE_EXISTING_FILE Code: AA 2922

Description: An attempt was made to add a disk file to the pool of backup files for the log. However, the disk file name specified on the ADD_FILE parameter already exists and contains data and will not be overwritten.

User Action: Specify a different backup file name on the ADD_FILE parameter.

--ERROR AA 2924-- File {file} was opened with logging turned on. This is the first update since the open. This non-fatal error does not concern the update, but informs you that BACKUP_PERMANENT_FILE must be done on this file before logging can actually happen. Meanwhile updates will continue, but without logging. You should back up the file before the next time it is opened, and then logging will begin.

Condition Identifier: AA\$LOGGING_NEEDS_BACPF Code: AA 2924

Description: None of the records you are adding will be logged until the indicated file is backed up.

User Action: Back up your keyed file (using the Backup_Permanent_File utility) before it is opened the next time; logging of your updates will begin.

--ERROR AA 2925-- {text} expects a NAME value for the {text} portion of the {text} parameter.

Condition Identifier: AA\$NAME_VALUE_EXPECTED Code: AA 2925

Description: The parameter value should be an SCL name.

User Action: Correct the parameter value. An SCL name begins with a nondigit and can contain letters, digits, and these special characters:

@ _ \$ [] { } \ | ~ `

--ERROR AA 2926-- File {text} : {text} was unable to {text} because exclusive access to this file was not possible.

Condition Identifier: AA\$CANT_GET_EXCLUSIVE_ACCESS Code: AA 2926

Description: Exclusive access to the file you specified was not possible.

User Action: Check that you have correct access to the file.

--ERROR AA 2927-- The file was not restored because Restore_Permanent_File returned the following error:

Condition Identifier: AA\$RESTORE_UNSUCCESSFUL Code: AA 2927

Description: The RESTORE_PERMANENT_FILE command was not successful.

User Action: None.

--ERROR AA 2928-- File {file} : When the file to be restored does not currently exist the cycle specification is required as part of the file path supplied to Recover_Keyed_File.

Condition Identifier: AA\$CYCLE_NUMBER_REQUIRED Code: AA 2928

Description: If the file being restored does not already exist, you must specify the cycle as part of the file path, with the RECOVER_KEYED_FILE Utility.

User Action: Specify the cycle on the file path.

--ERROR AA 2929-- {text}: {text} encountered a {text} system error.

Condition Identifier: AA\$SYSTEM_ERROR_IN_UTILITY Code: AA 2929

Description: This is an internal error.

User Action: Report this to your system analyst.

--ERROR AA 2930-- {text} expects a FILE value for the {text} portion of the {text} parameter.

Condition Identifier: AA\$FILE_VALUE_EXPECTED Code: AA 2930

Description: The parameter value must specify a file. A file is specified by a file path consisting of one or more names separated by periods (name.name.name). The names defines the path by which the file entry is found in the catalog hierarchy. The last name in the path is the file name. A cycle reference and/or an open position designator can be appended to the file path.

User Action: Correct or remove the parameter value.

--ERROR AA 2931-- {text} requires the log control file to exist if the RETAIN_CONFIGURATION parameter value is TRUE.

Condition Identifier: AA\$RETAIN_CONFIG_REQUIRES_LOG Code: AA 2931

Description: The configuration of the log cannot be retained because the absence of the log control file means that the configuration is incomplete.

User Action: You must delete the log including the configuration, that is, DELETE_LOG RETAIN_CONFIGURATION=FALSE, then recreate the log. Or, you can restore a backup copy of the log control file, if you have backups, using the RESTORE_LOG_CONTROL_FILE command.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2932-- Log {text} : {text} cannot be performed on a log that is currently in the process of being created.

Condition Identifier: AA\$CMD_ILLEGAL_FOR_NEW_LOG Code: AA 2932

Description: Task cannot be performed on a log that is currently in the process of being updated.

User Action: Wait until new log has completed.

--ERROR AA 2933-- File {text} cannot be used for a log backup file because {text}.

Condition Identifier: AA\$CANT_USE_FILE_FOR_LOG_BKUP Code: AA 2933

Description: This file cannot be used for a log backup file.

User Action: Specify a valid file for the log backup.

--ERROR AA 2934-- File {text}: {text} does not yet support this form of update to the log control file.

Condition Identifier: AA\$CANT_UPDATE_LOG_CNTL_FILE Code: AA 2934

Description: This form of update to the log control file is not supported.

User Action: Check with your site administrator on supported utilities using the log control file.

--ERROR AA 2935-- {text} cannot delete the nested_file because it is currently selected.

Condition Identifier: AA\$CANT_DELETE_NESTED_FILE Code: AA 2935

Description: The specified nested file cannot be deleted because it is currently selected.

User Action: Select another nested file (preferably \$MAIN_FILE because it cannot be deleted).

Further Information: See the AMP\$SELECT_NESTED_FILE description in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2936-- {text} requires that at least 3 repositories must exist when log backups are specified.

Condition Identifier: AA\$NOT_ENOUGH_REPOSITORIES Code: AA 2936

Description: Only two repositories and no backup files are configured for the log.

User Action: Either specify at least two backup files (one for each repository) or increase the number of repositories for the log.

Further Information: See the CONFIGURE_LOG_BACKUP or CONFIGURE_LOG_RESIDENCE subcommand description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2937-- Log {text} : Access to this log requires a minimum user privilege of ring {text}.

Condition Identifier: AA\$NOT_PRIVILEGED_FOR_LOG Code: AA 2937

Description: Access to this log does require a minimum user privilege of ring.

User Action: Notify the Log Administrator so he can give you access.

--ERROR AA 2938-- {text} is not permitted in the same session with {text}.

Condition Identifier: AA\$ILLEGAL_SUBCOMMAND Code: AA 2938

Description: A subcommand was used in the same session as another subcommand in which the two conflict.

User Action: Use one or the other subcommands.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2939-- Catalog {text} : {text} could not add the backup entry {text} because the tape file name may not be a path.

Condition Identifier: AAE\$TAPE_FILE_BKUP_MUST_BE_NAME Code: AA 2939

Description: An attempt was made to add a tape file to the pool of backup files for the log. However, the tape file name specified on the ADD_FILE parameter must be a local name, not a path.

User Action: Specify a local name on the ADD_FILE parameter.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2940-- {text} requires that the file specified for the {text} parameter be a sequential file.

Condition Identifier: AAE\$FILE_MAY_NOT_BE_KEYED_FILE Code: AA 2940

Description: The file must be a sequential file; it cannot be a keyed file.

User Action: Check that the correct file is specified.

--ERROR AA 2941-- {text} requires that if log backups are specified, the number of backups must be equal to the number of repositories. There are {text} repositories and {text} backup entries.

Condition Identifier: AAE\$INCORRECT_COUNT_LOG_BKUPS Code: AA 2941

Description: If you specify log backups, you must specify a backup file for each repository that the log is configured to use.

User Action: Specify a CONFIGURE_LOG_BACKUP subcommand for each repository.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2942-- Catalog {text} : {text} could not add the backup entry {text} because the backup file may not be a temporary file.

Condition Identifier: AAE\$ADD_FILE_MAY_NOT_BE_LOCAL Code: AA 2942

Description: The subcommand specified a backup file that is a temporary mass storage file.

User Action: Change the backup file to a permanent mass storage file or a magnetic tape file.

--ERROR AA 2943-- Catalog {text} : {text} can not delete the currently selected log.

Condition Identifier: AAE\$CANT_DELETE_CURRENT_CATALOG Code: AA 2943

Description: A log cannot be deleted after the session has specified it as the log to be modified by the session.

User Action: If the session is to delete and then reconfigure a log, the DELETE_LOG subcommand must precede the CONFIGURE_LOG_RESIDENCE subcommand.

--ERROR AA 2944-- Catalog {text} : {text} could not add the backup entry {text} because a backup entry already exists for that file.

Condition Identifier: AAE\$BACKUP_ENTRY_EXISTS Code: AA 2944

Description: The specified backup file could not be added to the backup pool because it already is in the pool.

User Action: Check that the correct file name or path is specified on the ADD_FILE parameter. Display the current backup pool by entering a DISPLAY_LOG_CONFIGURATION subcommand.

Further Information: See the CONFIGURE_LOG_BACKUP and DISPLAY_LOG_CONFIGURATION subcommand descriptions in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2945-- {text} found a value for the {text} parameter that is not allowed because {text}.

Condition Identifier: AAES\$INVALID_PARM_VALUE Code: AA 2945

Description: The parameter value is invalid in this context.

User Action: Correct the parameter value. To go to the command description in the AFM manual, enter HELP,command.

Further Information: The command and its parameters are described in the NOS/VE Advanced File Management manual.

--ERROR AA 2946-- {text} does not allow the keyword value {text} for {text}.

Condition Identifier: AAES\$ILLEGAL_KEYWORD_FOR_PARM Code: AA 2946

Description: The specified keyword is invalid in this context.

User Action: Correct the invalid keyword. To see the valid keywords for the command, enter: DISPLAY_COMMAND_PARAMETERS,command or enter HELP,command to go to the command description in the AFM manual.

Further Information: For further information on a command, see the command description in the NOS/VE Advanced File Management manual.

--ERROR AA 2947-- The {text} subcommand was entered and an ADD_FILE was specified, but the job accounting information for the log backup jobs has not been provided. Please enter the Set_Log_Backup_Account subcommand, and then enter QUIT again.

Condition Identifier: AAES\$ADD_BKUP_REQUIRES_ACCOUNT Code: AA 2947

Description: When a backup pool is specified for a log, validation information for backup jobs must also be specified in the log configuration.

User Action: Enter a SET_LOG_BACKUP_ACCOUNT subcommand specifying the validation information to be used by the LOGIN statement of each backup job for the log.

Further Information: See the SET_LOG_BACKUP_ACCOUNT subcommand description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2948-- The FILE_CONTENTS attribute of the log file {text} was not {text}, as required.

Condition Identifier: AAES\$BAD_FILE_CONTENTS_FOR_LOG Code: AA 2948

Description: The FILE_CONTENTS attribute of the log file has been changed to an inappropriate value.

User Action: Change the FILE_CONTENTS attribute value to the value specified in the message using a CHANGE_FILE_ATTRIBUTES command. (You can display the attribute value using a DISPLAY_FILE_ATTRIBUTES command.)

Further Information: See the CHANGE_FILE_ATTRIBUTES and DISPLAY_FILE_ATTRIBUTES command descriptions in the NOS/VE Commands and Functions manual.

--ERROR AA 2949-- Log {text} : {text} cannot retain the configuration of this log because the log is incomplete.'

Condition Identifier: AAES\$CANT_RESET_LOG Code: AA 2949

Description: The configuration of the log cannot be retained because the absence of a repository file means that the configuration is incomplete.

User Action: You must delete the log including the configuration, that is, DELETE_LOG RETAIN_CONFIGURATION=FALSE, then recreate the log. Or, you can restore a backup copy of the repository file, if you have backups, using the RESTORE_LOG command.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2950-- {text} requires that both values for the {text} parameter be of the same type.

Condition Identifier: AAE\$KEY_TYPE_MISMATCH Code: AA 2950

Description: The indicated parameter values are not the same kind of value.

User Action: Correct the parameter specification. An SCL value set is enclosed in parentheses and a space or comma must separate the values in the value set.

Further Information: See the command description in the NOS/VE Advanced File Management Usage manual or in the NOS/VE Commands and Functions manual.

--ERROR AA 2951-- {text} requires that backups be configured for the log. The Configure_Log_Backup subcommand must be entered at the time the log is created to configure the log backup files.

Condition Identifier: AAE\$CMD_REQUIRES_BACKUP_ENTRIES Code: AA 2951

Description: To restore a log or log control file, the log must have backups.

User Action: If log backups are desired, the log must be configured for log backups in the same session in which the log is created.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2952-- {text} was entered with CATALOG={file}. The current catalog is {text}. The Administer_Recovery_Log utility does not allow the catalog to be changed.

Condition Identifier: AAE\$CATALOG_MAY_NOT_BE_CHANGED Code: AA 2952

Description: Each Administer_Recovery_Log session can configure only one log.

User Action: If the first log specified in the session is not to be configured, enter a CANCEL_LOG_CHANGES subcommand; you can then specify another catalog or a CONFIGURE_LOG_RESIDENCE subcommand. Or, if the configuration specified for the first log is to be carried out, enter a QUIT subcommand.

Further Information: See the NOS/VE Advanced File Management Usage manual for the Administer_Recovery_Log utility description.

--ERROR AA 2953-- Repository {text} : {text} was unable to reinitialize this repository because {text}. This log is NOT usable. It must be deleted, and then it may be recreated.

Condition Identifier: AAE\$CANT_RESET_REPOSITORY Code: AA 2953

Description: The configuration cannot be retained because the specified repository is damaged or inaccessible.

User Action: You must delete the log and its configuration (DELETE_LOG RETAIN_CONFIGURATION=FALSE), then recreate the log.

Further Information: See the NOS/VE Advanced File Management usage manual.

--ERROR AA 2955-- {text} cannot create the requested direct-access file because when a loading_factor value is specified, the dynamic_home_block_space value must be TRUE.

Condition Identifier: AAE\$LOADING_FACTOR_WITHOUT_DHBS Code: AA 2955

Description: The dynamic home block space feature is not yet implemented.

User Action: Remove the loading_factor value or specify FALSE for the dynamic_home_block_space value and 0 as the loading_factor value.

--ERROR AA 2956-- Catalog {text} : {text} does not allow a \$STATION catalog to be used for a log.

Condition Identifier: AAE\$ILLEGAL_CATALOG_PATH Code: AA 2956

Description: The specified catalog is a \$STATION catalog.

User Action: Specify another catalog on the subcommand.

--ERROR AA 2957-- {text} found a value for {text} that is not permitted because {text}.

Condition Identifier: AA\$ILLEGAL_VALUE_FOR_PARM Code: AA 2957

Description: The specified parameter value is not allowed.

User Action: Correct the parameter value.

Further Information: See the NOS/VE Advanced File Management Usage manual for the subcommand description.

--ERROR AA 2958-- File {text} does not have the required ring attributes (4, 4, 4) for a log file.

Condition Identifier: AA\$LOG_FILE_HAS_INVALID_RINGS Code: AA 2958

Description: The RING_ATTRIBUTES value for one of the log files is wrong. This probably means that the file was not created using the Administer_Recovery_Log utility.

User Action: 1. Backup all keyed files that use the log. 2. Execute an Administer_Recovery_Log session to delete and then reconfigure the log.

Further Information: See the Administer_Recovery_Log description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2959-- {text} requires that a value be specified for either the ADD_FILE or REMOVE_FILE parameters, or both.

Condition Identifier: AA\$MUST_SPECIFY_ADD_OR_REMOVE Code: AA 2959

Description: You must specify a file name on the subcommand. If the subcommand is only to change the specifications for a file, you should specify the name of the file on both the ADD_FILE and REMOVE_FILE parameters.

User Action: Specify the file on the ADD_FILE and/or REMOVE_FILE parameters.

Further Information: See the CONFIGURE_LOG_BACKUP subcommand description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2960-- {text} does not allow the {text} parameter to be specified for the {text} file organization.

Condition Identifier: AA\$PARAM_INVALID_FOR_FILE_ORG Code: AA 2960

Description: The specified parameter is invalid for the given file organization.

User Action: Remove the invalid parameter.

Further Information: See the keyed-file utilities description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2961-- File {text} : Permission not granted for {text} .

Condition Identifier: AA\$RING_VALIDATION_ERROR Code: AA 2961

Description: You do not have the proper ring permissions to access the specified file.

User Action: None.

--ERROR AA 2962-- {text} requires that the {text} be supplied when {text}.

Condition Identifier: AA\$REQUIRED_PARM_NOT_SUPPLIED Code: AA 2962

Description: The request could not be carried out because a required parameter is missing.

User Action: Specify a value for the required parameter.

Further Information: See the subcommand description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2963-- Catalog {text} : {text} is not permitted after the log has been created.

Condition Identifier: AAE\$SETVL_ILLEGAL_FOR_OLD_LOG Code: AA 2963

Description: The subcommand is valid only when creating a new log, not when modifying an existing log.

User Action: To change the verification level, you must recreate the log. To do so, first backup all files that use the log and then execute an Administer_Recovery_Log session to delete the log and then re-create it with the desired verification level.

Further Information: See the Administer_Recovery_Log description in the NOS/VE Advanced File Management manual.

--ERROR AA 2964-- {text} is not allowed because this instance of open has a non-exclusive file lock set.

Condition Identifier: AAE\$NOT_WHILE_FILE_LOCKED Code: AA 2964

Description: This instance of OPEN has a file lock set.

User Action: Notify the Log Administrator.

--ERROR AA 2965-- {text} cannot use file {text} as both the input file and the output file.

Condition Identifier: AAE\$SAME_INPUT_AND_OUTPUT_FILE Code: AA 2965

Description: The same file cycle is specified as both the input file and the output file.

User Action: Change either the input file or output file specification. If appropriate, specify cycle references on the input and/or output file references to indicate that the files are different cycles of the same file. For example, specifying \$NEXT as the cycle reference for the output file creates the output file as the next higher cycle of the input file.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2966-- Recover_Keyed_File was unable to open the log file for file {text} and no backup entries were found for this file in the system log. Recovery was aborted.

Condition Identifier: AAE\$UNABLE_TO_ACCESS_USER_LOG Code: AA 2966

Description: The RECOVERULT_DESCRIPTION_TEXT and no backup entries were found for this file in the system log. Recovery was aborted.

User Action: Notify the Log Administrator for assistance.

--ERROR AA 2967-- File{text} {file}{text} : The {text} request was issued, but a parcel is in progress for this file identifier. This request is not allowed within the boundaries of a parcel.

Condition Identifier: AAE\$ILLEGAL_WITHIN_PARCEL Code: AA 2967

Description: The request cannot be performed within a parcel.

User Action: Move the request outside of the parcel. If the call is to lock the file, the file lock must be released before the parcel can begin.

--ERROR AA 2968-- File{text} {file}{text} : The {text} request was issued, but the file is open for read access. Parcels are designed for files having write access.

Condition Identifier: AAE\$READ_ONLY_PARCEL Code: AA 2968

Description: It is not appropriate to begin a parcel to update a file when the task has not opened the file for updating.

User Action: Check that the correct file is specified. If the file is to be updated, change the open request so it requests access modes to update the file. If the file is not to be updated, remove the parcel calls for the file.

--ERROR AA 2969-- {text} may only be used after a catalog has been established with the Use_Log subcommand.

Condition Identifier: AAESUPDATE_WITH_NO_CURRENT_LOG Code: AA 2969

Description: The subcommand is not valid until the catalog for the log to be created or modified has been specified by a CONFIGURE_LOG_RESIDENCE subcommand.

User Action: Enter a CONFIGURE_LOG_RESIDENCE subcommand specifying at least the CATALOG parameter.

Further Information: See the CONFIGURE_LOG_RESIDENCE subcommand description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2970-- {text} requires that parameter {text} have the single value ALL or a list of {text}.

Condition Identifier: AAESALL_NOT_USED_ALONE Code: AA 2970

Description: If specified, the keyword value ALL must be the only value for a parameter. It cannot be specified with other values.

User Action: Either remove the other parameter values and specify only ALL or remove the ALL keyword value.

--ERROR AA 2971-- File{text} {file}{text} : The {text} request was issued, but no parcel was currently active for the file identifier specified.

Condition Identifier: AAESPARCEL_NOT_FOUND Code: AA 2971

Description: The program attempted to end a parcel for a file identifier for which no parcel is in effect.

User Action: Check that the file identifier specified on a preceding begin_parcel call is the same file identifier specified on the call to commit or abort the parcel. The status returned by each update call in the parcel should be checked to determine whether the system has aborted the parcel.

Further Information: See the Parcels discussion in the CYBIL Keyed-File and Sort/Merge Interfaces or the FORTRAN Language Definition manual.

--ERROR AA 2972-- File{text} {file}{text} : The {text} request was issued, but the limit of file parcels active for this task has been reached. This parcel will not be started. No cleanup is necessary.

Condition Identifier: AAESTOO_MANY_PARCELS Code: AA 2972

Description: The task cannot begin another parcel until it commits or aborts one of its existing parcels.

User Action: Rewrite the program so it commits or aborts its parcels as soon as possible.

Further Information: See the Parcels discussion in the CYBIL Keyed-File and Sort/Merge Interfaces or the FORTRAN Language Definition manual.

--ERROR AA 2973-- File{text} {file}{text} : The {text} request was issued, but the file attributes include AMC\$ENABLE_MEDIA_RECOVERY, and space for the records to be logged cannot be allocated. This parcel will not be started. No cleanup is necessary.

Condition Identifier: AAESSPACE_NOT_AVAILABLE Code: AA 2973

Description: The request was issued, but space for the records cannot be allocated. Parcel will not be started and no cleanup is necessary.

User Action: Make sure enough space is available and retry.

--ERROR AA 2974-- File{text} {file}{text} : The {text} request was issued, but the parcel could not be committed successfully. This parcel was aborted and will require no additional cleanup.

Condition Identifier: AAESPARCEL_ABORTED Code: AA 2974

Description: The attempt to commit the parcel failed.

User Action: Retry the parcel.

--ERROR AA 2975-- File{text} {file}{text} : {text} found a record in which a key delimiter character was missing for a variable-length alternate key. No value for that key was taken from that record.

Condition Identifier: AAE\$MISSING_KEY_DELIMITER Code: AA 2975

Description: An attempt to update the alternate index for a variable-length alternate key found a key value too long for the alternate key. No key value can exceed the key length specified when the alternate key was defined. While collecting key values from the record, no delimiter character was found within key_length+1 bytes.

User Action: Check the contents of the record to determine if the data is correct. If necessary, delete the existing variable-length key and redefine the alternate-key definition so that the appropriate key values are included in the key index.

Further Information: Variable-length alternate keys and alternate-key creation are described in the CYBIL Keyed-File & Sort/Merge Interfaces manual, the FORTRAN Usage manual, and the NOS/VE Advanced File Management manual.

--ERROR AA 2976-- The {text} subcommand may not be used when {text}.

Condition Identifier: AAE\$ILLEGAL_SUBUTILITY_COMMAND Code: AA 2976

Description: The subcommand is invalid in this context.

User Action: To see a list of the valid subcommands, enter the subcommand HELP.

--ERROR AA 2977-- AMP\$GET_NEXT_RESULT_SET_RECORD was requested indicating the data file{text} {file}{text}. This request requires that \$PRIMARY_KEY be selected and that the nested file selected be {text}.

Condition Identifier: AAE\$ILLEGAL_KEY_OR_NF_SELECTED Code: AA 2977

Description: An attempt to read records using a result set failed due to one or both of the following reasons:

- An alternate key is selected instead of the primary key. (The primary key must be selected because the key values in the result set are primary-key values.)
- The nested file specified on the read request does not match the nested file for which the result set was created.

User Action: Check that the intended result set is specified on the read request.

Check the nested file selection. If no nested file has been explicitly selected, the default nested file, \$MAIN_FILE, is selected. The selected nested file must be the same nested file selected while the result set was built.

Check the key selection for the nested file. (The key selection must follow the nested file selection.) If an alternate key was selected, select the primary key (\$PRIMARY_KEY) before attempting to read records using a result set.

Further Information: See the Result Sets discussion in the FORTRAN Usage manual or the CYBIL Keyed-File & Sort/Merge Interfaces manual. For a FORTRAN program, the result set read request is RSGETN and the nested file and key are selected by storing their names in the FIT with STOREF calls. For a CYBIL program, the result set read request is AMP\$GET_NEXT_RESULT_SET_RECORD and the nested file and key are selected by AMP\$SELECT_NESTED_FILE and AMP\$SELECT_KEY calls, respectively.

--ERROR AA 2978-- File{text} {file}{text} : The {text} request was issued, but a parcel is already in progress for this file identifier. Only one parcel may exist for each open file in the task.

Condition Identifier: AAE\$PARCEL_IN_PROGRESS Code: AA 2978

Description: The task could not begin the parcel because it already has a parcel in progress for the same instance of open.

User Action: Finish each parcel for an instance of open before beginning the next parcel for the same instance of open. If more than one parcel in progress is required for the same file, use more than one instance of open.

--ERROR AA 2979-- File{text} {file}{text} : The {text} request was issued, but a parcel cannot be started because the logging options for this file do not include amc\$enable_parcel.

Condition Identifier: AA\$PARCEL_NOT_ALLOWED Code: AA 2979

Description: The task attempted to begin a parcel to update a file whose LOGGING_OPTIONS attribute does not include the option ENABLE_PARCELS.

User Action: Check that the correct file is being used. If so, the file owner should change the LOGGING_OPTIONS attribute of the file using a CHANGE_FILE_ATTRIBUTES command.

--ERROR AA 2980-- File{text} {file}{text} : {text} failed because the current alternate key has variable length, but the specified key length is 0.

Condition Identifier: AA\$ZERO_LENGTH_KEY Code: AA 2980

Description: The request specified the key value length as 0 bytes. When using a variable-length alternate key to read or position a file, you must specify the actual key value length as the major_key_length value.

User Action: Specify the key value length in bytes as the major_key_length (mkl) parameter value.

Further Information: The major_key_length parameter for the CYBIL calls AMP\$GET_KEY and AMP\$START is described in the CYBIL Keyed-File & Sort/Merge Interfaces manual. The mkl parameter for the FORTRAN calls GET and STARTM is described in the FORTRAN Usage manual.

--ERROR AA 2981-- File {file} : {text} does not accept the continuation file for input because the last operation completed successfully.

Condition Identifier: AA\$CONTINUE_NOT_ALLOWED Code: AA 2981

Description: The continuation file is available only when the previous command failed due to a record error. A message is issued when this occurs.

User Action: Do not enter a command that specified AAF\$CONTINUE when no message has been received indicating that the file is available.

Further Information: For further information on the ADD_RECORDS, COMBINE_RECORDS, and REPLACE_RECORDS subcommands, see the NOS/VE Advanced File Management manual.

--ERROR AA 2982-- File {file} : {text} cannot copy the input file to the currently selected nested file because the input file contains more than one nested file, and the input nested file was not specified.

Condition Identifier: AA\$CANT_COPY_SEVERAL_NFS_TO_NF Code: AA 2982

Description: The output file specification included a nested-file name, but the input file specification did not and the input file contains more than one nested file. The command can copy only one nested file, but it does not know which nested file to copy.

User Action: Add a nested-file name to the input file specification (such as \$MAIN_FILE for the default nested file) or remove the nested-file name from the output file specification.

A nested file is specified by enclosing the two values (the file reference and the nested-file name) in parentheses. The values must be separated by a comma or a space.

Further Information: For more information on copying nested files, see the NOS/VE Advanced File Management manual.

--ERROR AA 2983-- File {file} : {text} does not contain a record with the key {text} or that starts with {text}.

Condition Identifier: AA\$MAJOR_KEY_NOT_FOUND Code: AA 2983

Description: The command searched for a record in the file whose primary-key value matches the specified value or begins with the specified value, but no such record exists in the file.

User Action: Check that the correct key value was specified.

Further Information: For further information on specifying key values, see the command description in the NOS/VE Advanced File Management manual.

--ERROR AA 2984-- File {file} : {text} does not contain a record with the key {text}.

Condition Identifier: AAE\$SPECIFIED_KEY_NOT_FOUND Code: AA 2984

Description: The command searched for a record with the specified primary-key value, but the currently selected nested file does not contain such a record.

User Action: Check that the correct key value was specified.

Further Information: For further information on specifying key values, see the command description in the NOS/VE Advanced File Management manual.

--ERROR AA 2985-- File {text} {file}{text} : {text} can't be done because variable length in an alternate-key definition is incompatible with FIFO handling of duplicate key values, concatenation, null suppression, and sparse keys. At least one of these exists in the key definition.

Condition Identifier: AAE\$NOT_WITH_VARIABLE_LENGTH Code: AA 2985

Description: The alternate-key definition specifies a variable-length key and at least one of the following:

- Duplicate key values are allowed, but the key lists are not sorted (First-in,first-out key lists).
- One or more pieces are to be concatenated to form the key value.
- Null alternate-key values are not stored in the alternate index.
- Alternate-key values are included or excluded based on a sparse-key control character.

User Action: Redefine the alternate key without the incompatible attributes and apply the definition to the file.

--ERROR AA 2986-- {text} detected two or more records with the same primary key value in the input file(s). No records have been applied to the output file. The list of duplicate records has been saved in file {text}. Use DISPLAY_FILE rather than COPY_FILE to display {text} on a terminal if the output file is a direct_access file, or if the records may contain non_printable ASCII characters or binary or integer information.

Condition Identifier: AAE\$DUPLICATE_KEY_IN_SORT Code: AA 2986

Description: The command found two or more input records with the same primary-key value. (If the primary key is collated, the values were found to be identical after collation.) The command wrote the duplicate records to the file given in the message. It processed none of the input records.

User Action: Look at the contents of the duplicate records file. Determine if the records are in error or the primary-key definition should be changed for the file.

Further Information: For more information on the primary-key definition and the CREATE_KEYED_FILE or CHANGE_KEYED_FILE subcommands, see the NOS/VE Advanced File Management manual.

--ERROR AA 2987-- No records were found in the input file(s) specified for {text}.

Condition Identifier: AAE\$NO_RECORDS_TO_COPY Code: AA 2987

Description: The specified input file or files are empty.

User Action: Check that the correct files were specified and that their open position is not \$EOI.

--ERROR AA 2988-- File {file} : {text} does not allow a list of files when the CREATE_KEYED_FILE recovery file is specified.

Condition Identifier: AAE\$CONTINUE_FILE_IN_LIST Code: AA 2988

Description: When the continuation file is specified, only the continuation file can be specified on the command.

User Action: Re-enter the command, specifying only the AAF\$CONTINUE file.

--ERROR AA 2989-- File {file} : The CREATE_KEYED_FILE continuation file does not exist.

Condition Identifier: AA\$CONTINUE_FILE_DOESNT_EXIST Code: AA 2989

Description: The continuation file (AAF\$CONTINUE) does not exist and so should not be referenced. The continuation file exists only when a record error has interrupted the processing of the previous command; a message is issued prompting the user to use the continuation file.

User Action: None required.

--ERROR AA 2990-- File {text} in {file}: {text} will not merge two nested files when their embedded_key attribute values are not the same. Files with the same embedded_key attribute value may be merged, or a file may be copied to a new file with a different embedded_key attribute value.

Condition Identifier: AA\$MERGE_WITH_EMB_KEY_CHG Code: AA 2990

Description: While copying records into an existing keyed file, the operation found a nested file that it could not copy because its EMBEDDED_KEY attribute differs from the EMBEDDED_KEY attributes of the nested file with the same name in the output file. To be merged, nested files with the same name must both have embedded keys or both have nonembedded keys. The other nested files were merged successfully.

User Action: None required. If appropriate, copy the uncopied nested file to a temporary file and then copy the temporary file to a new nested file in the output file. If the embedded key is at key position 0, the temporary file can be copied to the existing nested file with the same name because, in that case, the primary-key values will not change when the EMBEDDED_KEY attributes differ.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--ERROR AA 2991-- File {file}: {text} encountered a sort error of severity {text}.

Condition Identifier: AA\$CREKF_SORT_ERROR Code: AA 2991

Description: The command called Sort/Merge to sort records. Sort/Merge returned an error of the indicated severity.

User Action: Try to determine why the sort failed. Refer to the Sort/Merge error message.

--ERROR AA 2992-- File {file} : {text} requires a value for the {text} file attribute before creating this file.

Condition Identifier: AA\$FILE_ATTRIBUTE_NOT_PRESET Code: AA 2992

Description: The indicated attribute has not been defined for the file.

User Action: Specify the attribute using a SET_FILE_ATTRIBUTES command.

Further Information: The complete SET_FILE_ATTRIBUTES command format is in the NOS/VE Commands and Functions manual. The NOS/VE Advanced File Management Usage manual describes the keyed-file attributes.

--ERROR AA 2993-- File {file} : {text} found a record in this file that is shorter than the minimum_record_length of the output file. This file will be ignored.

Condition Identifier: AA\$RECORD_SHORTER_THAN_MINRL Code: AA 2993

Description: Because the input file contains a record shorter than the number of bytes specified by the minimum_record_length attribute of the output file, none of the records from the input file have been placed in the output file.

User Action: Check that the correct input file was specified. If so, then check that the minimum_record_length value for the output file is appropriate. (Use the command DISFA,file,DO=MINRL.) If appropriate, edit the input file to remove or lengthen the short record.

Further Information: For more information on an appropriate minimum record length, see the NOS/VE Advanced File Management manual.

--ERROR AA 2994-- File {file} : {text} failed because of a lock held by another task on the lock as a whole (file_freeze lock).

Condition Identifier: AAE\$LOCKED_OUT_OF_FILE Code: AA 2994

Description: The operation could not be performed because another task has a lock on the entire file that prevents the requested operation.

User Action: Retry the operation later. If appropriate, specify waiting on the next request for the operation so that the request waits until the file is available (or the request times out).

Further Information: For more information on lock processing, see the CYBIL Keyed-File & Sort/Merge Interfaces manual or the FORTRAN Usage manual.

--ERROR AA 2995-- File {file} : {text} expects an executable procedure as the ERROR_EXIT_PROCEDURE value. The value provided was not in an executable section.

Condition Identifier: AAE\$BAD_ERROR_EXIT_PROCEDURE Code: AA 2995

Description: In a CYBIL program, the value for the ERROR_EXIT_PROCEDURE attribute must be specified as a pointer to the procedure. In a FORTRAN program, the ERROR_EXIT_PROCEDURE value must be a name (not enclosed in apostrophes) declared as EXTERNAL in the program.

(This differs from the ERROR_EXIT_PROCEDURE_NAME attribute which is specified as an entry point name to be loaded. The ERROR_EXIT_PROCEDURE value must specify a procedure that is already loaded because it changes the procedure in use after the file is opened.)

User Action: Either correct the ERROR_EXIT_PROCEDURE value or specify the error processing procedure before the file is opened with the ERROR_EXIT_PRCcEDURE_NAME attribute.

Further Information: See the error exit procedure attribute descriptions in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--ERROR AA 2996-- File{text} {file}{text} : {text} failed because an invalid name {text}. A name must begin with a nondigit and contain only letters, digits, and the special characters _, @, #, \$, [,], \, ^.

Condition Identifier: AAE\$INVALID_NAME Code: AA 2996

Description: The specified name must follow the SCL naming rules. It must be from 1 to 31 characters, begin with a nondigit, and contain only letters, digits, and these special characters:

_ @ # \$ [] { } \ | ~ ^

User Action: Specify a valid SCL name.

--ERROR AA 2997-- File{text} {file}{text} : {text} failed because KA was not given as 0 when KEY_NAME = ALL.

Condition Identifier: AAE\$ALL_SPECIFIED_AND_KA_NOT_0 Code: AA 2997

Description: When the specified KEY_NAME is ALL, the key_address (KA) value must be zero. In other words, the request must specify that the operation is to be performed on ALL or on a single key value.

User Action: Correct the KEY_NAME or KEY_ADDRESS specification.

--ERROR AA 2998-- File {file} : {text} requires that the values for the input and output file attributes be the same. The output file attribute {text} has the value {text}, which does not match the input file value.

Condition Identifier: AAE\$ILLEGAL_FILE_ATTRIBUTE Code: AA 2998

Description: One of the file attributes do not match for the input and output files and so the operation cannot be performed.

User Action: If possible, change the output file attribute so it matches the input file attribute.

Further Information: The complete SET_FILE_ATTRIBUTES command format is in the NOS/VE Commands and Functions manual. The NOS/VE Advanced File Management manual describes the keyed-file attributes.

--FATAL AA 3000-- No FIT parameter given on {text} request - required on all FORTRAN calls to the keyed-file interface.

Condition Identifier: AAES\$NO_FIT_PARAMETER_GIVEN Code: AA 3000

Description: The first parameter value on a FORTRAN keyed-file interface call must be a variable containing a pointer to a FIT.

User Action: Specify the fit parameter for the given operation.

Further Information: The FIT parameter and the FORTRAN keyed-file interface calls are described in the FORTRAN Language Definition Usage manual.

--FATAL AA 3005-- Invalid FIT parameter given on {text} request - FILEIS or FILEDA call must precede other FORTRAN calls to the keyed-file interface, and the FIT must remain intact.

Condition Identifier: AAES\$INVALID_FIT Code: AA 3005

Description: The fit parameter value must be the variable containing the FIT pointer returned by the OPENM call that opened the file. A FILEIS call must precede other FORTRAN keyed-file interface calls, and the file information table (FIT) must be kept intact.

User Action: Check that FILEIS is called and correct the fit parameter specification.

Further Information: See the description of the specified call under Keyed-File Interface Calls in the FORTRAN Language Definition Usage manual.

--FATAL AA 3010-- File{text} {file}{text} : {text} called with too many parameters.

Condition Identifier: AAES\$TOO_MANY_PARAMETERS_GIVEN Code: AA 3010

Description: The call specifies too many parameter values.

User Action: Remove the excess parameter values.

Further Information: See the call descriptions in the FORTRAN Language Definition Usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3015-- File{text} {file}{text} : {text} requires more parameters.

Condition Identifier: AAES\$NOT_ENOUGH_PARAMETERS Code: AA 3015

Description: The call specified too many parameter values.

User Action: Refer to the call description for the correct number of parameter values.

Further Information: The FORTRAN keyed-file interface calls are described in the FORTRAN Language Definition Usage manual.

--FATAL AA 3020-- File{text} {file}{text} : {text} called missing a keyword or a value.

Condition Identifier: AAES\$PARAMETER_KEYWORD_MISMATCH Code: AA 3020

Description: A keyword in the call does not have a matching value, or a value does not have a matching keyword.

User Action: Be sure that there is a keyword for every value in the call.

Further Information: The FIT keywords and values are described in the FORTRAN Language Definition Usage manual.

--FATAL AA 3025-- {text} requires that a local file name (LFN) be previously specified.

Condition Identifier: AAES\$LFN_REQUIRED_TO_OPEN Code: AA 3025

Description: The file name of the keyed file must be specified in order to open the file.

User Action: Specify the file name. Only the file name is required, not the complete file reference. The system searches for the file name in the \$LOCAL catalog; thus, if the file is permanent, it must be attached.

Further Information: To read about attaching a file, see the NOS/VE System Usage manual. To read about an open request, see the FORTRAN Language Definition Usage manual (OPENM call), the COBOL Usage manual (OPEN statement), or the CYBIL File Management manual (FSP\$OPEN_FILE call).

--FATAL AA 3030-- File{text} {file}{text} : {text} attempted for a NEW file on an OLD file.

Condition Identifier: AAES\$CANT_OPEN_NEW_AN_OLD_FILE Code: AA 3030

Description: The operation attempted to access the file as a new file, but the system considers the file to be an old file because it has been previously opened.

User Action: Check the open request for the file.

Further Information: To read about the FORTRAN OPENM call, see the FORTRAN Language Definition Usage manual. To read about COBOL indexed sequential file opens, see the COBOL Usage manual. To read about the CYBIL AMP\$OPEN call, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3035-- File{text} {file}{text} : {text} not allowed on a sequential or byte addressable file.

Condition Identifier: AAES\$FILE_NOT_INDEXED_SEQUENTIAL Code: AA 3035

Description: The operation was attempted on a file that does not have indexed- sequential file organization.

User Action: Check that the correct file was specified. To display the FILE_ORGANIZATION attribute, specify the file on this command:

```
DISPLAY_FILE_ATTRIBUTE, file, ..  
DISPLAY_OPTION=FILE_ORGANIZATION
```

For a new file, redefine the file, specifying the appropriate keyed-file attributes.

--FATAL AA 3040-- File{text} {file}{text} : {text} attempted for an OLD file on a NEW file.

Condition Identifier: AAES\$CANT_OPEN_OLD_A_NEW_FILE Code: AA 3040

Description: The operation attempted to access a new file (a file that has not been previously opened) as an old file (a file that has been previously opened).

User Action: Check the open request for the file. Also, check that, if the file is a permanent file, it was attached before the open request.

Further Information: To read about the SCL ATTACH_FILE command, see the NOS/VE System Usage manual. To read about the FORTRAN OPENM call, see the FORTRAN Language Definition Usage manual. To read about the COBOL OPEN statement, see the COBOL Usage manual. To read about the CYBIL PFP\$ATTACH and FSP\$OPEN_FILE calls, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3045-- File{text} {file}{text} : {text} encountered an internal error (number {text})

Condition Identifier: AAES\$INTERFACE_INTERNAL_ERROR Code: AA 3045

Description: Internal error; file interface malfunction.

User Action: Follow your site procedures for reporting the error.

--FATAL AA 3050-- File{text} {file}{text} : {text} required space on the system heap in the user ring, but none was available.

Condition Identifier: AAES\$USER_RING_SYSTEM_HEAP_FULL Code: AA 3050

Description: Internal error; the operation was unable to acquire the needed system heap space.

User Action: Report the software error.

--FATAL AA 3200-- File{text} {file}{text} : {text} failed because the file was not closed after its last alteration. Attempt to re-create the file using COPY_KEYED_FILE.

Condition Identifier: AAES\$ALTERED_NOT_CLOSED Code: AA 3200

Description: Internal error; file recovery is required.

User Action: Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3205-- File{text} {file}{text} : {text} found an invalid name in the collate_table_name field while attempting to validate it. All fields in the nested_file_definition must be defined; this includes the collate_table_name field even if the key_type is not collated.

Condition Identifier: AAES\$COLLATE_TABLE_NAME_MISSING Code: AA 3205

Description: The collate_table_name field of the nested-file definition was not initialized. The collate_table_name must be initialized even if the key_type is not collated.

User Action: Initialize the collate_table_name field. If the key_type is not collated, the collate_table_name may be initialized as OSC\$NULL_NAME.

Further Information: See the AMP\$CREATE_NESTED_FILE call description in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL AA 3210-- File{text} {file}{text} : {text} detected an invalid MAXIMUM_RECORD_LENGTH of 0 or undefined.

Condition Identifier: AAES\$MAX_REC_LEN_0_OR_UNDEF Code: AA 3210

Description: The current MAXIMUM_RECORD_LENGTH (either zero or undefined) is not valid for the specified call.

User Action: Define the MAXIMUM_RECORD_LENGTH attribute as the maximum length, in bytes, of the data records.

Further Information: To read about the minimum and maximum record length attributes, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3215-- File{text} {file}{text} : {text} detected an invalid KEY_LENGTH of {text} which is greater than one word with KEY_TYPE = INTEGER_KEY.

Condition Identifier: AAES\$INTEGER_KEY_GT_ONE_WORD Code: AA 3215

Description: The key type is specified as integer, but its length is specified as longer than 8 bytes (1 word).

User Action: Correct either the key type or the key length.

Further Information: For a description of alternate-key definition, see the SCL Advanced File Management manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3220-- File{text} {file}{text} : {text} detected an invalid MAXIMUM_RECORD_LENGTH - greater than {text}.

Condition Identifier: AAES\$MAX_REC_LENGTH_TOO_BIG Code: AA 3220

Description: The specified maximum record length is greater than 65497 bytes, the maximum value allowed for keyed files.

User Action: Specify a maximum record length from 1 through 65497.

Further Information: To read about the maximum_record_length attribute, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual (BLOCK CONTAINS Clause for Indexed Sequential Files), the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3225-- File{text} {file}{text} : {text} detected an invalid MINIMUM_RECORD_LENGTH - it must be less than or equal to MAXIMUM_RECORD_LENGTH.

Condition Identifier: AAES\$MIN_GT_MAX_REC_LENGTH Code: AA 3225

Description: The minimum record length for a file must be less than the maximum record length.

User Action: Decrease the minimum record length attribute value or increase the maximum record length attribute value. The minimum and maximum record length values can be the same.

Further Information: See the description of the MAXIMUM_RECORD_LENGTH and MINIMUM_RECORD_LENGTH attributes in the COBOL Usage, FORTRAN Language Definition, CYBIL Keyed-File and Sort/Merge Interfaces Usage, or SCL Advanced File Management Usage manual.

--FATAL AA 3230-- File{text} {file}{text} : {text} detected {text} too small for KEY_POSITION + KEY_LENGTH - key must fit in the minimum record length.

Condition Identifier: AAE\$REC_TOO_SMALL_FOR_KEY Code: AA 3230

Description: The key field must occur within the minimum record length.

User Action: Check that the key position and key length values are correct and the minimum record length attribute is correct.

Further Information: To read about the key_length, key_position, and minimum_record_length attributes, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3235-- File{text} {file}{text} : {text} detected incompatible DATA_PADDING and MAXIMUM_RECORD_LENGTH values - padding must allow one maximum-length record in a block.

Condition Identifier: AAE\$DATA_PAD_TOO_LARGE Code: AA 3235

Description: The data block padding percentage must be small enough to allow at least one maximum-length record in a block.

User Action: Change the values of the data padding and/or maximum record length attributes as necessary.

Further Information: To read about data block padding and the data_padding attribute, see the NOS/VE Advanced File Management manual, the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual. To see the complete SET_FILE_ATTRIBUTES command format, refer to the NOS/VE Commands and Functions manual.

--FATAL AA 3240-- File{text} {file}{text} : {text} detected incompatible INDEX_PADDING and KEY_LENGTH values - padding must allow 3 index records in a block.

Condition Identifier: AAE\$INDEX_PAD_TOO_LARGE Code: AA 3240

Description: The index_padding percentage is too large. It must allow at least three index records (of length KEY_LENGTH + 4 bytes) in an index block.

User Action: Decrease the index_padding attribute value.

Further Information: To read about the INDEX_PADDING attribute, see the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3245-- File{text} {file}{text} : {text} cannot be done without {text} permission.

Condition Identifier: AAE\$AAM_REQUIRES_ACCESS Code: AA 3245

Description: To perform the requested operation, the file must be opened for the indicated access mode. If the file is attached before it is used, it must be attached with the indicated access mode in its usage mode set. The applicable file permit must grant the requested access to the user.

User Action: Check that the indicated access modes are requested when the file is attached and opened.

If the file belongs to another user, check that you have been granted the required access by displaying the file permits with the command:

DISPLAY_CATALOG_ENTRY, file, DISPLAY_OPTION=PERMITS

Further Information:

- SCL file permits display: See the DISPLAY_CATALOG_ENTRY command description in the SCL System Interface Usage manual.
- SCL access mode specification: See the SET_FILE_ATTRIBUTES or ATTACH_FILE command descriptions in the SCL System Interface Usage manual.
- FORTRAN access mode specification: See the OPENNM call description in the FORTRAN Language Definition Usage manual.
- COBOL access mode specification: See the indexed-sequential file open description in the COBOL Usage manual.
- CYBIL access mode specification: See the access_mode attribute description and the AMP\$FILE and AMP\$OPEN call descriptions in the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3250-- File{text} {file}{text} : {text} failed because the ruined flag is set for the file. The flag was set previously when a structural error was detected. If enable_media_recovery is among the values for the Logging_Options file attribute, the file may be restored with the Recover_Keyed_File utility. Otherwise, attempt to re-create the file using COPY_KEYED_FILE.

Condition Identifier: AA\$FILE_IS_RUINED Code: AA 3250

Description: The operation was denied because an earlier operation discovered a structural error in the file. File recovery is required.

User Action: Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3255-- File{text} {file}{text} : {text} : the user-declared maximum number of nonfatal errors has been reached since the last OPEN.

Condition Identifier: AA\$ERROR_LIMIT_EXCEEDED Code: AA 3255

Description: The fatal error was issued because the nonfatal (trivial) error limit (specified by the ERROR_LIMIT attribute or parameter) has been reached. Messages describing the nonfatal errors have been written to the \$ERRORS file (by default, the OUTPUT file).

User Action: Correct the errors and/or increase the ERROR_LIMIT value.

Further Information: To read about the ERROR_LIMIT attribute, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual.

To read about the ERROR_LIMIT parameter on the APPLY_KEY_DEFINITIONS or QUIT subcommands, see the SCL Advanced File Management manual.

--FATAL AA 3260-- File{text} {file}{text} : a physical read operation has returned an abnormal status so the file must be considered ruined. If you have write access, the ruin flag has been set. File recovery is required.

Condition Identifier: AA\$BAD_READ Code: AA 3260

Description: Internal error; file recovery required.

User Action: Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3265-- File{text} {file}{text} : the key type for this new file is given as COLLATED, but the COLLATE_TABLE_NAME attribute is nil or undefined, hence the file cannot be opened.

Condition Identifier: AA\$COLLATED_KEY_NEEDS_TABLE Code: AA 3265

Description: The primary-key type is defined as collated, but no collation table has been provided so the file cannot be opened.

User Action: Specify the collation table name or location as a file attribute and check that the collation table is accessible from an object library in the current object library list. To display the current object library list, enter a DISPLAY_PROGRAM_ATTRIBUTES command; to add an object library to the list, enter a SET_PROGRAM_ATTRIBUTES command; to display the contents of an object library, enter a DISPLAY_OBJECT_LIBRARY command..

Further Information: To read about program attribute and object library concepts and commands, see the NOS/VE Object Code Management manual. To read how to create and specify a collation table, see the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3270-- File {file} : {text} encountered system error while {text}.

Condition Identifier: AA\$SYSTEM_ERROR_OCCURRED Code: AA 3270

Description: A system error was found by the operation. The user is given a description of the situation.

User Action: Follow your site procedures for reporting the software error.

--FATAL AA 3275-- File{text} {file}{text} : {text} detected KEY_LENGTH not specified or specified as 0.

Condition Identifier: AAE\$KEY_LENGTH_0_OR_UNDEF Code: AA 3275

Description: The operation requires specification of a nonzero key length.

User Action: Specify the key length as an integer greater than zero.

--FATAL AA 3280-- File{text} {file}{text} : an internal procedure, AAP\$PROCESS_ABNORMAL_CONDITION, was called with an improper condition code.

Condition Identifier: AAE\$IMPROPER_CONDITION_CODE Code: AA 3280

Description: Improper use of a file interface routine was detected. Occurrence of this error usually indicates improper use of the file interface.

User Action: Follow the your site procedures to report the software error.

--FATAL AA 3285-- File{text} {file}{text} : {text} encountered an internal error (number {text}) that was probably caused by internal file structure damage. File recovery is required.

Condition Identifier: AAE\$PRODUCT_INTERNAL_ERROR Code: AA 3285

Description: Internal error; file recovery required. The internal error was probably caused by an error in the internal file structure.

User Action: Follow your site procedures for reporting the software error. Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3286-- File {file}{text} : {text} failed because the file staple could not be allocated in the lock_file_heap.

Condition Identifier: AAE\$ALLOCATE_FILE_STAPLE_FAILED Code: AA 3286

Description: This is a system internal error.

User Action: Report this to your site analyst.

--FATAL AA 3290-- File{text} {file}{text} : {text} by alternate key consulted a list of alternate-key values and extracted a primary-key value for which there is no record. File structure is not consistent so recovery is required.

Condition Identifier: AAE\$INVALID_PKV_FROM_MIP_FILE Code: AA 3290

Description: Internal error; file recovery required. The alternate-key operation, using a list of alternate-key values extracted a primary-key value for which there is no corresponding data record. This indicates that the file structure is in error.

User Action: Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3295-- File{text} {file}{text} : {text} found a link mismatch between records in a SYSTEM_KEY file. A placeholder points to an alien in another block, and an alien points back to the corresponding placeholder. Following one of these pointers, we have found either a record of the wrong type at the other end or a record of the right type, whose pointer doesn't point back to the starting record. File structure is disturbed so recovery required.

Condition Identifier: AAE\$BAD_S_K_OVERFLOW_LINK Code: AA 3295

Description: Internal error; file recovery is required. The system found a link mismatch between records in an system key file. A placeholder points to an alien in another block, and an alien points back to the corresponding placeholder. Following one of these pointers, we have found either a record of the wrong type at the other end, or a record of the right type, whose pointer does not point back to the starting record.

User Action: Report the error.

Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3300-- File{text} {file}{text} : {text} can't be done because the given maximum record size of {text} is too big to fit into a block of this file. Block size was fixed when the file was created so when a nested file is created, it has to use the same block size.

Condition Identifier: AAE\$MAX_REC_EXCEEDS_BLOCK Code: AA 3300

Description: The operation cannot be done because the given maximum record length is longer than the block length for this file. The block length is determined when the file is created; when a nested file is created, it must use the same block length.

User Action: Decrease the maximum record length to a value not greater than the block length. To see the current maximum record length and block length values, enter a DISPLAY_KEYED_FILE_PROPERTIES command.

Further Information: The NOS/VE Advanced File Management manual describes the DISPLAY_KEYED_FILE_PROPERTIES command. To read about the maximum_record_length and maximum_block_length attributes, see the FORTRAN Language Definition Usage manual, the COBOL Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3305-- File{text} {file}{text} : {text} can't be done for a DIRECT_ACCESS file or nested file because neither an initial_home_block_count nor dynamic_hashing (not yet released) was specified.

Condition Identifier: AAE\$NO_HOME_BLOCK_COUNT Code: AA 3305

Description: A direct-access file structure cannot be created without an initial_home_block_count specification. (The dynamic_hashing feature is not yet available.)

User Action: Specify a positive integer as the INITIAL_HOME_BLOCK_COUNT attribute value.

A FORTRAN or CYBIL program can specify the attribute value from within the program; a COBOL program cannot. In that case, the value must be specified by this command before the file is created:

```
SET_FILE_ATTRIBUTE, file, INITIAL_HOME_BLOCK_COUNT=integer
```

Further Information: For a description of how to choose a home block count for a direct-access file, see the CYBIL Keyed-File and Sort/Merge Interfaces manual or the NOS/VE Advanced File Management Usage or Tutorial manual.

--FATAL AA 3310-- File{text} {file}{text} : {text} cannot be done because the named hashing procedure cannot be loaded.

Condition Identifier: AAE\$CANT_LOAD_HASH_ROUTINE Code: AA 3310

Description: The entry point name specified by the HASHING_PROCEDURE_NAME attribute cannot be found in the program library list.

User Action: Check that the correct hashing procedure name has been specified. Add the object library containing the hashing procedure module to the program library list using the command:

```
SET_PROGRAM_ATTRIBUTES ADD_LIBRARY=file
```

If appropriate, use the default hashing procedure, AMP\$SYSTEM_HASHING_PROCEDURE (no object library necessary).

Further Information: See the HASHING_PROCEDURE_NAME attribute description in the NOS/VE Advanced File Management Usage, FORTRAN Usage, or CYBIL Keyed-File and Sort/Merge Interfaces manual. See the program library list and SET_PROGRAM_ATTRIBUTES command descriptions in the NOS/VE Object Code Management manual.

--FATAL AA 3315-- File{text} {file}{text} : the {text} feature is not supported at this level.

Condition Identifier: AAE\$UNIMPLEMENTED_FEATURE Code: AA 3315

Description: The request attempted to use a feature not implemented for this software release.

User Action: Change the request so that it does not attempt to use the unimplemented feature.

Further Information: See the description of keyed-file features in the appropriate manual (the FORTRAN Language Definition Usage manual, the COBOL Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual, or the NOS/VE Advanced File Management Usage manual).

--FATAL AA 3320-- File{text} {file}{text} :file must be considered ruined because a forced write returned an abnormal status. The file-ruined flag has been set.

Condition Identifier: AA\$BAD_WRITE Code: AA 3320

Description: Internal error; file recovery required.

User Action: Report the software error.

Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3325-- File{text} {file}{text} : {text} detected a user-specified MINIMUM_RECORD_LENGTH not equal to the MAXIMUM_RECORD_LENGTH. For fixed-length records the MINIMUM must equal the MAXIMUM. Change the record_length values to match or change the record_type.

Condition Identifier: AA\$MIN_NE_MAX_ON_FIXED_RECS Code: AA 3325

Description: When the record type is ANSI fixed-length (F), the minimum record length and maximum record length attribute values must be the same.

This error can occur if the file was created by a COPY_KEYED_FILE operation when the EMBEDDED_KEY attributes of the input and output files differed. Because the operation adds or removes the primary-key value from the data, it changes the minimum record length value. Thus, the minimum record length no longer matches the maximum record length.

User Action: Change either the minimum record length or the maximum record length attribute value so that the values are the same.

Further Information: To read about the minimum and maximum record length attributes, see the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, the COBOL Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--FATAL AA 3330-- File{text} {file}{text} : write-modified-pages during flush returned bad status. Ruined flag has been set for the file.

Condition Identifier: AA\$BAD_FLUSH Code: AA 3330

Description: Internal error; file recovery required.

User Action: Request a reload of the archived file copy. Attempt to update the reloaded file to its state when the error occurred.

--FATAL AA 3335-- {text} requires the second value, the nested_file name, on parameter {text} to be an SCL NAME.

Condition Identifier: AA\$BAD_FILE_PARAMETER Code: AA 3335

Description: The parameter value must specify a file.

User Action: Correct the parameter value.

--FATAL AA 3340-- {text} expected NAME value for the nested_file_name value on parameter {text}.

Condition Identifier: AA\$BAD_NESTED_FILE_PARAMETER Code: AA 3340

Description: The second value in the parameter value set must be a NOS/VE name. A NOS/VE name begins with a nondigit and contains only letters, digits, and the characters # _ \$ @ [] { } \ | ~ ^ .

User Action: Correct or remove the second value in the value set.

--FATAL AA 3345-- File{text} {file}{text} : {text} encountered an internal error (number {text}) in command logic or internal file structure damage. File recovery may be required.

Condition Identifier: AA\$COMMAND_INTERNAL_ERROR Code: AA 3345

Description: Internal error; the command encountered an error in its logic or internal file structure damage.

User Action: Enter a DISPLAY_KEYED_FILE_PROPERTIES command to determine if the file has a structural error. Report the software error if appropriate. Reload the file if necessary.

--FATAL AA 3350-- {text} requires parameter {text}.

Condition Identifier: AAE\$REQUIRED_PARAMETER Code: AA 3350

Description: You must specify a value for the indicated parameter on the command.

User Action: Specify a valid value for the parameter.

Further Information: The keyed-file commands are described in the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3355-- {text} requires {file} to be an existing file.

Condition Identifier: AAE\$FILE_EXPECTED_TO_EXIST Code: AA 3355

Description: The operation requires that the file be an existing file, that is, a file that has been previously opened.

User Action: Check that the correct file was specified.

--FATAL AA 3365-- File {file} : {text} cannot process a file with file organization = {text}.

Condition Identifier: AAE\$KEYED_FILE_EXPECTED Code: AA 3365

Description: The file must be a keyed file, that is, its file_organization attribute must be a keyed-file organization.

User Action: Check that the correct file is specified. If appropriate, copy the file to a new keyed file using the COPY_KEYED_FILE command.

--FATAL AA 3370-- File{text} {file}{text} : {text} cannot be done because the named compression/decompression procedure cannot be loaded.

Condition Identifier: AAE\$CANT_LOAD_COMPRESS_ROUTINE Code: AA 3370

Description: The entry point name specified by the COMPRESSION_PROCEDURE_NAME attribute is not in the program library list.

User Action: Check that the correct name is specified as the COMPRESSION_PROCEDURE_NAME attribute. The attribute is optional. To clear the attribute value, specify a null string ("") as the attribute value.

To add the object library file containing the user-written compression procedure to the program library list, you can use this command:

```
SET_PROGRAM_ATTRIBUTES, ADD_LIBRARY=file
```

Further Information: For more information on user-written compression procedures, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3375-- File {file} : {text} is not allowed with BYTE_ADDRESSABLE file_organization.

Condition Identifier: AAE\$BYTE_ADDRESSABLE_ILLEGAL Code: AA 3375

Description: The file organization attribute of a COPY_KEYED_FILE input or output file cannot be byte-addressable.

User Action: If the file is a new file, change its file organization to sequential. If the file is an old file, use the COPY_FILE command to copy the file to a new sequential file.

Further Information: To see the SET_FILE_ATTRIBUTES and COPY_FILE command descriptions, refer to the NOS/VE System Usage manual.

--FATAL AA 3380-- File {file} : {text} requires that the input file (or nested file) exist, that is, it must have been opened in the past.

Condition Identifier: AAE\$EMPTY_INPUT_FILE Code: AA 3380

Description: COPY_KEYED_FILE checks that the input file is an existing file, not a new file. A new file cannot contain data to be copied because it has never been opened.

User Action: Check that the correct input file is specified.

Further Information: To see the COPY_KEYED_FILE command description, refer to the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3385-- File {file} : {text} requires enough information in the input file to create a new nested file. A sequential file is not sufficient. To create a new nested file, the input must either name a nested file or be of a file_organization which uses keys.

Condition Identifier: AAES\$CANT_COPY_SQ_TO_NEW_NF Code: AA 3385

Description: To create a new nested file, COPY_KEYED_FILE must copy keyed-file information from the input file. Thus, the input file must be a keyed file or a nested file in a keyed file.

User Action: Check that the correct files are specified. If appropriate, create the new nested file before the COPY_KEYED_FILE command.

--FATAL AA 3390-- File {file} : {text} found several nested files, but the output is singular (either a nested file or a file organization not supporting nested files, such as sequential).

Condition Identifier: AAES\$CANT_COPY_SEVERAL_NFS_TO_1 Code: AA 3390

Description: Only one nested file can be copied to another nested file or to a file with sequential file organization.

User Action: Either specify a single nested file to be copied or change the output file specification so that it either omits the nested file name or specifies a keyed file.

--FATAL AA 3395-- File {file} : {text} detected an attempt to copy a file or nested file to itself. The input and output files or nested files must differ.

Condition Identifier: AAES\$CANT_COPY_TO_SELF Code: AA 3395

Description: The COPY_KEYED_FILE command specified the same file cycle or the same nested file as both input and output.

User Action: Correct either the input or output file specification.

Further Information: To see the COPY_KEYED_FILE command description, refer to the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3400-- File{text} {file}{text} : {text} found that the FILE_CONTENTS attribute value is LIST, which is invalid for a keyed file. FILE_CONTENTS may be any other value.

Condition Identifier: AAES\$FILE_CONTENTS_ILLEGAL Code: AA 3400

Description: The FILE_CONTENTS attribute of a keyed file cannot be LIST. This attribute value is not appropriate because keyed file cannot be a listing file. (The first character in each record of a listing file is assumed to be a format effector [carriage control] character.)

User Action: Check that the correct file was specified. If so, change the FILE_CONTENTS attribute of the file to another value (such as LEGIBLE).

Further Information: See the CHANGE_FILE_ATTRIBUTE command description in the NOS/VE Commands and Functions or the NOS/VE System Usage manual.

--FATAL AA 3405-- File {file} : {text} does not allow the same file to be used for both input and output.

Condition Identifier: AAES\$INPUT_FILE_SAME_AS_OUTPUT Code: AA 3405

Description: The input and output files cannot be the same file cycle.

User Action: Correct the input file or output file specification.

--FATAL AA 3410-- File {file} : file has been ruined as requested.

Condition Identifier: AAES\$APPLY_IS_TERMINATED Code: AA 3410

Description: The user entered RUIN FILE in response to the prompt sent after the user entered a terminate_break_character that interrupted an apply operation.

User Action: The file is no longer usable. If appropriate, create a new file to replace the ruined file.

Further Information: For a description of apply interrupt processing, see the Remarks in the APPLY_KEY_DEFINITIONS subcommand description in the NOS/VE Advanced File Management manual.

--FATAL AA 3415-- {text} required space on the system heap in the user ring for {text}, but none was available.

Condition Identifier: AA\$CMD_SYS_HEAP_USER_RING_FULL Code: AA 3415

Description: Internal error; the system was unable to allocate space on the system heap in the ring in which the user is executing.

User Action: Report the software error.

--FATAL AA 3420-- File {file} : {text} cannot create the file specified by the {text} parameter because the file already exists and contains data.

Condition Identifier: AA\$FILE_ALREADY_EXISTS Code: AA 3420

Description: When attempting to create the file, the request found another file with the same name already existing and containing data.

User Action: Check that the correct file is specified. If the existing file is an attached permanent file and a temporary file is to be created, detach the attached file. If necessary and appropriate, delete the existing file.

Further Information: For the DETACH_FILE and DELETE_FILE command descriptions, see the NOS/VE System Usage manual.

--FATAL AA 3425-- File {file} : Because there were no input records, the value of the INITIAL_HOME_BLOCK_COUNT could not be calculated. {text} cannot create the direct_access output file.

Condition Identifier: AA\$NO_VALUE_FOR_IHBC Code: AA 3425

Description: The QUIT command was entered in response to an IHBC/ prompt, but no input records were available to calculate the initial_home_block_count value for the file. Creation of a new direct-access file requires specification of an initial_home_block_count value so the file was not created.

User Action: Specify the initial_home_block_count attribute value with the other attributes for the new direct-access file using a IHBC parameter on a SET_FILE_ATTRIBUTES command before the CREATE_KEYED_FILE command.

Otherwise, enter ADD_RECORDS commands in response to the IHBC/ prompts. The ADD_RECORDS commands must specify all input files that contain input records for the new direct-access file.

Further Information: For more information, see the descriptions of direct-access file structure and the CREATE_KEYED_FILE utility in the NOS/VE Advanced File Management manual.

--FATAL AA 3430-- File{text} {file}{text} : {text} operation could not be done because the number of blocks in the file has reached, or has approached, the FILE_LIMIT; thus, the file is now ruined.

Condition Identifier: AA\$FILE_AT_FILE_LIMIT Code: AA 3430

Description: The keyed-file structure has lost its integrity because its size has reached the FILE_LIMIT attribute value.

User Action: Reconstruct the file if possible. Do not set the FILE_LIMIT attribute; its default value is its maximum value.

Further Information: See the file structure attributes description in the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3435-- File{text} {file}{text} : {text} operation could not be done because too many locks exist for the file. While this is true, the only valid operation for the file is a CLOSE.

Condition Identifier: AA\$LOCK_FILE_CROWDED Code: AA 3435

Description: The lock manager can process no more locks for this file.

User Action: Close the file, releasing all locks held by the instance of open. If possible, change the program so it minimizes the number of locks required and releases locks as soon as possible. If the program requires several concurrent key value locks, it should request a single file lock instead of individual key value locks. If necessary, attach the file for exclusive access (share_modes=none) so that no locks are necessary.

Further Information: See the lock processing description in the keyed-file interface section of the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL AA 3440-- File {file} : {text} cannot clear and overwrite the contents of \$MAIN_FILE in the output file without deleting the file. A merge copy (file position not BOI) is allowed.

Condition Identifier: AAESCANT_REPLACE_MAIN_FILE Code: AA 3440

Description: The default nested file (\$MAIN_FILE) cannot be overwritten.

User Action: To merge the input records into \$MAIN_FILE, append the open_position designator \$EOI to the output file reference.

To create a new file containing the data from the input file in its \$MAIN_FILE nested file and the data from the other output nested files, copy the input file to a new file and then copy each output nested file to the new file.

Further Information: See the COPY_KEYED_FILE command description in the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3445-- The {text} parameter for the {text} command requires {text}.

Condition Identifier: AAEBAD_COMMAND_PARAMETER Code: AA 3445

Description: A command parameter is in error.

User Action: Correct the command parameter as indicated in the message.

Further Information: See the command description in the NOS/VE Advanced File Management Usage manual.

--FATAL AA 3450-- File{text} {file}{text} : {text} operation could not be done because the given file organization is not indexed_sequential, direct_access, or system_key.

Condition Identifier: AAEBAD_FILE_ORG Code: AA 3450

Description: The indicated operation can only be performed on a file whose FILE_ORGANIZATION attribute is one of the keyed-file organizations (indexed-sequential, direct-access, or system-key).

User Action: Check that the correct file was specified. To display the FILE_ORGANIZATION attribute, specify the file on this command:

```
DISPLAY_FILE_ATTRIBUTE, file, ..  
DISPLAY_OPTION=FILE_ORGANIZATION
```

For a new file, redefine the file, specifying the appropriate keyed-file attributes.

Further Information: To read about the keyed-file organizations and attributes, see the FORTRAN Usage, CYBIL Keyed-File and Sort/Merge, or NOS/VE Advanced File Management Usage manual.

--FATAL AA 3455-- File{text} {file}{text} : {text} cannot be done because the specified RECORD_TYPE attribute value differs from the record_type of the file.

Condition Identifier: AAESCANT_CHANGE_RECORD_TYPE Code: AA 3455

Description: The operation cannot change the RECORD_TYPE attribute.

User Action: Display the RECORD_TYPE attribute and specify the same value.

Further Information: See the DISPLAY_FILE_ATTRIBUTE and SET_FILE_ATTRIBUTE command descriptions in the NOS/VE Commands and Functions manual.

--FATAL AA 3460-- File{text} {file}{text} : {text} timed out while waiting for a required resource.

Condition Identifier: AAE\$DEPENDENCY_TIMEOUT_ERROR Code: AA 3460

Description: The system could not perform the operation because the wait time limit expired before it could get access to part of the file. The default wait time limit is 60 seconds; however, it can be changed using the SCL variable AAV\$RESOLVE_TIME_LIMIT.

User Action: Try again later.

If appropriate, display the value of the AAV\$RESOLVE_TIME_LIMIT variable, if it exists. A new value can be assigned to the variable by an SCL statement. The SCL statement can be executed from within a program. For example, the following statement can change the value from within a FORTRAN program:

```
Call SCLCMD('create_variable, name=AAV$RESOLVE_TIME_LIMIT,
+           kind=integer, value=45, scope=local')
```

Similarly, the following call can change the value from within a CYBIL program:

```
CLP$SCAN_COMMAND_LINE('create_variable, name=AAV$RESOLVE_TIME_LIMIT,
                       kind=integer, value=45, scope=local', status);
```

--FATAL AA 3465-- File {file} : {text} encountered system error while {text}.

Condition Identifier: AAE\$FTN_SYSTEM_ERROR_OCCURRED Code: AA 3465

Description: The FORTRAN keyed-file interface detected a system error.

User Action: Follow your site procedures for reporting problems.

--FATAL AA 3470-- File{text} {file}{text} : Cannot open file because share modes have not been defined.

Condition Identifier: AAE\$SHARE_MODES_UNDEFINED Code: AA 3470

Description: A share-mode set must be defined for the operation. The share mode set specifies the access modes that the instance of open allows other concurrent instances of open. For exclusive access to the file, specifies an empty share-mode set (share_modes=none).

User Action: To allow other instances of open to update the file, include the write modes in the share-mode set. To allow other instances of open to read the file, include read and modify modes in the share-mode set. To prevent sharing of the file, specify an empty share-mode set (share_modes=none).

If you allow other instances of open to update the file, you must include lock processing in your program to ensure the integrity of the file data.

Further Information: See the discussion of keyed-file sharing in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL AA 3475-- File{text} {file}{text} : Cannot open file because access and share modes conflict with those of existing instances of open.

Condition Identifier: AAE\$SHARE_MODE_CONFLICT Code: AA 3475

Description: An open request fails if either:

- Its access mode set is not included in the set of share modes allowed by all other instances of open of the file.
- Its share mode set does not include all access modes required by the other instances of open of the file.

User Action: Check that the open request specifies the minimal access mode set required and the maximum share mode set possible.

If appropriate, change the program so that it reissues the open request when it receives this status.

Further Information: See the ACCESS_MODE attribute and keyed-file sharing descriptions in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL AA 3480-- File {file} : {text} encountered an internal error: {text}. No recovery is necessary.

Condition Identifier: AAESCMD_ERROR_NO_RECOVERY_REQD Code: AA 3480

Description: An internal error was detected; however, the file structure is still intact so no recovery is required.

User Action: Report the software error.

--FATAL AA 3485-- File {file} : {text} encountered an internal error: {text}. Recovery may be required.

Condition Identifier: AAESCMD_ERROR_RECOVERY_REQD Code: AA 3485

Description: An internal error was detected. The error may have produced an error in the keyed-file structure.

User Action: Attempt to determine if the file is complete and correct. If appropriate, reload the file from the last archived copy and attempt to update it to the current state.

--FATAL AA 3490-- File {file} : {text} requires that the input file specified must currently exist and contain data.

Condition Identifier: AA\$INPUT_FILE_MUST_EXIST Code: AA 3490

Description: The specified input file does not exist or is empty.

User Action: Check that the correct input file was specified. If a new keyed file is to be created, use the CREATE_KEYED_FILE command.

Further Information: For more information on keyed-file creation, see the NOS/VE Advanced File Management manual.

--FATAL AA 3495-- File {file} : {text} requires that the attributes specified for the output file match with the attributes of the input file. The attributes that did not conform are listed in the \$ERRORS file.

Condition Identifier: AA\$FOUND_ILLEGAL_ATTRIBUTES Code: AA 3495

Description: The attributes of the output file must be the same as the attributes of the input file. The differing attributes have been written to file \$ERRORS.

User Action: Look at file \$ERRORS for the attributes in error. Check that the correct files were specified. If possible, change the file attributes or re-create one of the files so their attributes match.

Further Information: File attributes and file re-creation are described in the NOS/VE Advanced File Management manual.

--FATAL AA 3500-- File {file} : {text} failed because another task has a lock on the file that prevents it (called file_freeze lock.) This task waited 5 times the usual resolve_time_limit.

Condition Identifier: AA\$LOCKED_OUT_OF_OPEN Code: AA 3500

Description: The operation failed due to a lock on the file.

User Action: Try again later. If possible, attach the file for exclusive access (share_modes=none) so that the file is not shared while you use it.

Further Information: For more information on keyed-file sharing, see the FORTRAN Usage, COBOL Usage, or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3505-- {text} could not return to nested file {text} because of the following_error. Nested file {text} is currently selected.

Condition Identifier: AAESCANT_RETURN_TO_NESTED_FILE Code: AA 3505

Description: An error prevented the reselection of a nested file. The message gives the name of the nested file selected instead.

User Action: See the following message for the reason the nested file could not be selected and take the appropriate action.

--FATAL AA 3510-- File {file} : {text} does not allow the file specified on the {text} parameter to be {text}.

Condition Identifier: AA\$FILE_VALIDATION_ERROR Code: AA 3510

Description: The command found that the specified file has the wrong attribute.

User Action: Check that the correct file is specified. If possible, change the file attribute or create a new file with the correct attribute.

Further Information: For more information, see the description of creating and re-creating keyed files in the NOS/VE Advanced File Management manual.

--FATAL AA 3515-- File {file} : a block was locked by mmp\$lock_pages, but when the time to unlock it came, the image in real memory had apparently been replaced by an earlier image from disk. It looks as if NOS/VE had completely recovered the job, but discarded the wired-down page.

Condition Identifier: AA\$WIRED_BLOCK_OVERREAD Code: AA 3515

Description: Internal system error, indicating an incomplete job recovery. Changes to part of the file may be lost.

User Action: Consult a site analyst. If appropriate, reload the file from the last archive copy and attempt to update the file to its current state.

--FATAL AA 3520-- File {file} : Compression procedure {text}.

Condition Identifier: AA\$DECOMPRESSION_EXCEEDS_MAXRL Code: AA 3520

Description: The compression procedure returned a decompressed record longer than the maximum_record_length attribute for the file or longer than the working_storage_length specified on the read request.

User Action: If appropriate, increase the working storage length or the maximum_record_length. Determine if the compression procedure is correctly decompressing the data.

Further Information: For further information on writing compression procedures, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3525-- File {file} : Compression procedure {text}.

Condition Identifier: AA\$DECOMPRESSION_LT_MINRL Code: AA 3525

Description: The compression procedure returned a decompressed record shorter than the minimum_record_length attribute for the file.

User Action: Check that the minimum_record_length attribute is correct. Determine if the compression procedure is decompressing the data correctly.

Further Information: For further information on writing a compression procedure, see the NOS/VE Advanced File Management manual.

--FATAL AA 3530-- File {file} : Compression procedure {text}.

Condition Identifier: AA\$COMPRESSION_EXCEEDS_MAXRL Code: AA 3530

Description: The compression procedure returned a compressed record longer than the maximum_record_length attribute for the file.

User Action: Check that the maximum_record_length attribute is correct. Determine why the compression of the record increased its size.

Further Information: For more information on writing a compression procedure, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3535-- File {file} : {text} requires that the result set file be open.

Condition Identifier: AA\$FILE_NOT_OPEN Code: AA 3535

Description: The result set file has not been opened.

User Action: Open the result set file before attempting any operations using the result set.

Further Information: For more information on opening a result set, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3540-- {text} found that the data file definition in the source result set file {file} does not match the data file definition in the target result set file {file}. The request is not processed.

Condition Identifier: AAE\$DATA_FILES_DIFFER Code: AA 3540

Description: The source and target result set files must contain the same data file definition. The data file definition identifies the file cycle specified as the data file when the result set was first opened.

User Action: Check that the correct result sets were specified. If necessary, open a temporary result set to serve as the target result set.

Further Information: For more information on building and combining result sets, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3545-- {text} found that, although the data files match, the nested file ({text}) specified in the source result set file {file} does not match the nested file ({text}) specified in the target result set file {file}. The request is not processed.

Condition Identifier: AAE\$NESTED_FILES_DIFFER Code: AA 3545

Description: The source and target result set files must be for the same nested file in the file. The name of the currently selected nested file is stored in the result set file when the result set is first opened.

User Action: Check that the correct result sets were specified. If necessary, select the correct nested file and open a temporary result set to serve as the target result set.

Further Information: For more information on building and combining result sets, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3550-- File {file} : {text} was called to place the result in AMC\$RESULT_IN_TARGET, but the target file provided is the same as one of the input files. This request is not processed.

Condition Identifier: AAE\$TARGET_NOT_GIVEN Code: AA 3550

Description: The target result set file cannot be the same as either of the source result set files.

User Action: Check that the correct result set files are specified. If necessary, open a temporary result set file to serve as the target result set file.

Further Information: For more information on using result sets, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3555-- File {file} : {text} found that the data file definition specified on the request does not match the data file definition in the existing result set file. The request is not processed.

Condition Identifier: AAE\$WRONG_DATA_FILE Code: AA 3555

Description: The data file specified must be the same file cycle on all calls to open the result set.

User Action: Check that the correct data file cycle is specified.

--FATAL AA 3560-- File {file} : {text} found that although the data files match, the nested file specified on the request ({text}) does not match the nested file in the existing result set file ({text}).

Condition Identifier: AAE\$WRONG_NESTED_FILE Code: AA 3560

Description: The nested file currently selected is not the same nested file selected when the result set was first opened.

User Action: Check that the correct result set file is specified. Correct the nested file selection.

Further Information: For more information on selecting nested files and using result sets, see the FORTRAN Usage or CYBIL Keyed-Files & Sort/Merge Interfaces manual.

--FATAL AA 3565-- File {file} : {text} was called to open the result set file, but the associated data file {text} is empty, is not a keyed file, or does not exist.

Condition Identifier: AA\$FILE_MUST_EXIST Code: AA 3565

Description: The data file specified on the result set open request must be an existing keyed file that contains data.

User Action: Check that the correct data file is specified.

Further Information: For more information on opening a result set, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3570-- File {file} : a logical lock appears to be held by a task that the operating system no longer recognizes as active. AAM recovery will be able to recover from this situation when it is released. For now, close the file and try again later.

Condition Identifier: AA\$DEAD_TASK_HOLDS_LOCK Code: AA 3570

Description: Internal system error. A lock exists for a task that is no longer active.

User Action: Close the file if it is still open. Try again later. If the error reoccurs, consult a site analyst.

--FATAL AA 3575-- File {file} : AMP\$OPEN_RESULT_SET was called, but was unable to get space from the system heap to build the result set tables.

Condition Identifier: AA\$SYSTEM_HEAP_FULL Code: AA 3575

Description: The result set open operation could not allocate the temporary space it requires.

User Action: Try again later when the system is not so busy. If this error occurs frequently, consult a site analyst.

--FATAL AA 3580-- {text} was called, but the result set identifier supplied was not valid. The result set must be open for this call, and the result_set_id must not be modified by the caller.

Condition Identifier: AA\$INVALID_RESULT_SET_ID Code: AA 3580

Description: The specified result set id is not valid. The result set id is returned by the call that opens the result set; the id is valid until the result set is closed.

User Action: Check that the variable in which the result set id was returned is the variable used to specify the result set is on the other result set calls.

Further Information: For more information on opening result sets, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3585-- {text} encountered a fatal fault in a user (hashing or compression) routine. The actual fault will appear in your job log.

Condition Identifier: AA\$FAULT_IN_USER_ROUTINE Code: AA 3585

Description: The user-defined hashing procedure or compression procedure used with the file returned a fatal error.

User Action: Check that the file attribute values for HASHING_PROCEDURE_NAME and COMPRESSION_PROCEDURE_NAME are correct. If appropriate, you could change the attribute values to the system-provided procedures, AMP\$SYSTEM_HASHING_PROCEDURE and AMP\$RECORD_COMPRESSION, respectively.

Correct the error in the user-defined procedure before using it with any file.

Further Information: For more information on writing hashing procedures and compression procedures, see the CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3590-- File {file}: AMP\$GET_NEXT_RESULT_SET_RECORD was called with NOT = TRUE, but the file organization of the result set data file is {text}. This is invalid because the file cannot be read sequentially.

Condition Identifier: AA\$SIS_FILE_EXPECTED Code: AA 3590

Description: The result set NOT operator is valid only when the file is an indexed-sequential file. In other words, when the file is a direct-access or system-key file, you cannot use the result set to read all records from the file that are not in the result set.

User Action: Determine whether the request intended to read a record NOT in the result set. If not, change the NOT parameter to FALSE.

Further Information: For more information on reading using a result set, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3595-- File {file}: AMP\$GET_NEXT_RESULT_SET_RECORD was called, but the end of information file position has been returned on the previous call. The result set file must be repositioned before reading can continue.

Condition Identifier: AA\$REPEATED_READ_AT_EOI Code: AA 3595

Description: The result set is positioned at its end so no more records can be read using the result set until it is repositioned.

User Action: Change the program so that it checks for end-of-file position after each read using the result set.

Further Information: For more information on reading using a result set, see the FORTRAN Usage or CYBIL Keyed-File & Sort/Merge Interfaces manual.

--FATAL AA 3600-- File {file} has been ruined by the crash of a task that was accessing it. Recovery from backup and log is needed.

Condition Identifier: AA\$RUINED_IN_ABNORMAL_CLOSE Code: AA 3600

Description: Your file has been ruined by the crash of a task that was accessing it. Recovery from backup and log is needed.

User Action: Attempt to recover from backup.

--FATAL AA 3605-- File {file} must be closed and re-opened, after which processing can continue. Or if the current operation is an open, it should be re-tried after a short wait. If a parcel was in progress, it has been aborted.

Condition Identifier: AA\$FILE_MUST_BE_CLOSED Code: AA 3605

Description: The file specified must be closed and reopened before processing can continue. If your current operation is an open, retry again after a short wait. If parcel was in progress, the parcel was aborted.

User Action: Close the file and reopened it to continue processing.

--FATAL AA 3625-- File {text} has not been recovered successfully. It has been saved in file {text}. It must be restored before file {text} can be restored.

Condition Identifier: AA\$ABORTED_RECOVERY_DETECTED Code: AA 3625

Description: Unsuccessful file recovery. File must be restored before recovery is successful.

User Action: Restore aborted file, before attempting next recovery.

--FATAL AA 3640-- File{text} {file}{text} : {text} encountered an internal error. No directory entry was found for this file.

Condition Identifier: AA\$NO_DIRECTORY_ENTRY_FOR_FILE Code: AA 3640

Description: Encountered an internal error. No directory entry was found for this file.

User Action: Notify the Log Administrator.

--FATAL AA 3655-- File{text} {file}{text} : {text} failed because there wasn't enough room in the log to record the operation.

Condition Identifier: AAE\$NO_ROOM_FOR_UPDATE_IN_LOG Code: AA 3655

Description: file faEFAULT_DESCRIPTION_TEXT operation.d because there was not enough space in the log to record the

User Action: Supply enough space for new update to log.

--FATAL AA 3660-- File{text} {file}{text} : Any file having AMC\$ENABLE_PARCELS as one of the recovery options must include at least AMC\$MODIFY in the list of access modes. This AMP\$OPEN request failed.

Condition Identifier: AAE\$PARCEL_OPEN Code: AA 3660

Description: When attempting to open a keyed file whose LOGGING_OPTIONS attribute includes the option ENABLE_PARCELS, the open request did not specify MODIFY in the access mode set and specified additional share modes besides READ.

User Action: Correct the access modes to include MODIFY access or change the share modes to either READ only or NONE.

--FATAL AA 3665-- File {text} : Cycle 2 of the log control file already exists and could not be deleted. The log control file was not changed.

Condition Identifier: AAE\$LOG_CONTROL_FILE_2_EXISTS Code: AA 3665

Description: Cycle 2 of the log control file (AAF\$LOG_CONTROL_FILE) exists, indicating an interruption of a previous Administer_Recovery_Log session.

User Action: 1. Check whether cycle 1 of AAF\$LOG_CONTROL_FILE in the log catalog exists. To do so, you could list the cycles using this command: DISPLAY_CATALOG_ENTRY, .log_catalog.AAF\$LOG_CONTROL_FILE, DISPLAY_OPTION=CYCLES 2. If cycle 1 does not exist, copy cycle 2 to cycle 1, as follows: COPY_FILE, .log_catalog.AAF\$LOG_CONTROL_FILE.2, .log_catalog.AAF\$LOG_CONTROL_FILE.1 3. Delete cycle 2, as follows: DELETE_FILE, .log_catalog.AAF\$LOG_CONTROL_FILE.2

Further Information: See the NOS/VE System Usage manual for descriptions of displaying, copying, and deleting file cycles.

--FATAL AA 3670-- A terminate break has been received. No log updates have been performed.

Condition Identifier: AAE\$USER_TERMINATED_ADMRL Code: AA 3670

Description: A terminate break has been received. No log updates have been performed.

User Action: Restart the ADMINISTER_RECOVER_LOG when you are ready.

--FATAL AA 3675-- The Administer_Recovery_Log utility terminated with a fatal error: {text}.

Condition Identifier: AAE\$ADMRL_TERMINATION_ERROR Code: AA 3675

Description: The Administrator_Recovery_Log Utility has terminated with a fatal error.

User Action: Check the files the Administer_Recovery_Log was recovering.

--FATAL AA 3685-- File{text} {file}{text} : {text} can't be done for a DIRECT_ACCESS file or nested file because the proposed number of home blocks > max_blocks_in_file_value, i.e. 231-1. As a practical limit this is absurdly high, but it prevents range errors.**

Condition Identifier: AAE\$TOO_MANY_HOME_BLOCKS Code: AA 3685

Description: Operation cannot be done for a direct-access file or nested file because the proposed number of home blocks is greater than the max_blocks_in_file value, i.e 2**31-1.

User Action: Reduce the number of home blocks.

--FATAL AA 3690-- File {file} : {text} was interrupted by calling a condition handler for {text}. AAM has disposed of the condition and turned it into this fatal error.

Condition Identifier: AA\$CONDITION_HANDLED_BY_AAM Code: AA 3690

Description: This may be an internal error.

User Action: Report this to your system analyst.

--FATAL AA 3695-- A previous AAM operation was interrupted by job time limit. The operation was allowed to complete or backed out, so as to leave the file in good order, or the file was marked ruined, but the task is not allowed to do any more AAM operations except CLOSE.

Condition Identifier: AA\$TIME_LIMIT Code: AA 3695

Description: The current operation has been interrupted by job time limit. Before processing can continue on the file, it must be closed.

User Action: Close the file. In the future, if more time is needed for the job, set the job time limit higher.

--FATAL AA 3700-- A previous AAM operation was interrupted by interactive termination break. The operation was allowed to complete or backed out, so as to leave the file in good order, or the file was marked ruined, but the task is not allowed to do any more AAM operations except CLOSE.

Condition Identifier: AA\$INTERACTIVE_TERMINATION Code: AA 3700

Description: The current operation has been interrupted by a user terminate break. Before further processing can be done on the file, it must be closed.

User Action: Close the file. If you do not close the file and processing continues on the file, the file will automatically be closed for you.

--FATAL AA 3705-- File {file} : {text} was attempted, but the file_identifier received by AAM has a 0 ordinal, or an ordinal unknown to AAM; or the sequence field of the fid differs from the sequence AAM has previously seen with this ordinal.

Condition Identifier: AA\$IMPROPER_FILE_ID Code: AA 3705

Description: The file-identifier used in this operation is not valid. This suggests the file is not open.

User Action: Make sure the file is open with this file-identifier.

--FATAL AA 3710-- FILE {file} : this file has been recreated by RESTORE_PERMANENT_FILE, which may have given you an older version of the file than you expect. This makes the log unusable. You have 2 choices: a) if this was your intent and the current version of your file is satisfactory, use the VOID_LOG_FOR_RESTORED_FILE subcommand of RECOVER_KEYED_FILE and then backup the file with BACKUP_PERMANENT_FILE to restart logging, or b) if you think RESTORE_PERMANENT_FILE has lost updates, and you want to recover as many as possible from the log, use the RECOVER_FILE_MEDIA subcommand of RECOVER_KEYED_FILE.

Condition Identifier: AA\$FILE_HAS_BEEN_RESTORED Code: AA 3710

Description: Until choice a or b is made, you will continue to receive this message every time you try to open the file.

User Action: Select choice a or b.

--FATAL AA 3715-- FILE {file} : When the log for this file was created, a bad user-name and/or password were given with SET_LOG_BACKUP_ACCOUNT. This has now been detected while trying to spin off a batch job to back up a repository in the log. You must use the administer-log utility to delete the log and then re-create it with a valid user-name and password.

Condition Identifier: AAE\$LOG_BACKUP_VALIDATION_BAD Code: AA 3715

Description: To test the validation of the user and password accounting information, a batch job is executed. In this case, the test failed because the user-name and/or the password are invalid.

User Action: You must delete the log and recreate the log with the correct user-name and password on the SET_LOG_BACKUP_ACCOUNT subcommand.

--FATAL AA 3720-- A system error prevents sharing of keyed files. Please notify your system analyst; there is further information in the \$SYSTEM log.

Condition Identifier: AAE\$LOCK_FILE_CORRUPTED Code: AA 3720

Description: This is a system internal error.

User Action: Report this error to your system analyst.

--FATAL AA 3725-- {text} encountered the following internal error - {text}.

Condition Identifier: AAE\$SUBSYSTEM_INTERNAL_ERROR Code: AA 3725

Description: This is a system internal error.

User Action: Report this error to your system analyst.

--FATAL AA 3735-- File {file} : The nested file name designation {text}, is not permitted for a sequential file.

Condition Identifier: AAE\$NF_ILLEGAL_FOR_SEQ_FILE Code: AA 3735

Description: A sequential file cannot contain a nested file.

User Action: Check your command syntax to make sure a nested file is not implied.

--FATAL AA 3740-- It appears that you are using a version of AAM that is coded for a non-existent cycle of the lock file.

Condition Identifier: AAE\$LOCK_FILE_CYCLE_MISSING Code: AA 3740

Description: You tried to access a keyed file that is using a cycle of the default lock file \$SYSTEM.AAM.AAF\$LOCK_FILE that does not exist.

User Action: Use the Administer_Lock_File utility to create a new lock file for the keyed file, and connect the lock file and the keyed file.

Further Information: See the section Administer_Lock_File Utility in the Advanced File Management manual.

--FATAL AA 3745-- File {file} : operation failed because the relevant lock file is already in use and appears to be corrupted. This file must be closed.

Condition Identifier: AAE\$LOCK_FILE_DOUBTFUL Code: AA 3745

Description: The current lock file is open and invalid.

User Action: Close the lock file. You may need to create a new lock file for the keyed file with the Administer_Lock_File utility.

--FATAL AA 3750-- File {file} : OPEN failed because the relevant lock file does not look like a valid lock file, or is private to a keyed file other than the one being opened.

Condition Identifier: AAE\$LOCK_FILE_MISMATCH Code: AA 3750

Description: The lock file is not a valid lock file, or the lock file is private and is connected to a different keyed file than the one you specified.

User Action: Connect the keyed file to a different lock file, such as the default lock file \$SYSTEM.AAM.AAF\$LOCK_FILE. If you clear the connect with CLEAR_LOCK_FILE_CONNECTION, the keyed file is automatically connected to the default lock file.

--CATASTROPHIC AA 4000-- File{text} {file}{text} : {text} was attempted after a fatal error, but only AMP\$FLUSH and AMP\$CLOSE and attribute operations are allowed after a fatal error.

Condition Identifier: AAE\$OUTSTANDING_FATAL_ERROR Code: AA 4000

Description: Another operation was attempted after a fatal error occurred; only flush and close requests and attribute operations are allowed after a fatal error.

User Action: Change the program so that it checks for a fatal error before continuing.

Further Information: To read about error checking for calls, see the COBOL Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces Usage manual.

--INFORMATIVE AA 5000--

Condition Identifier: AAE\$INVALID_LOCK_INFO_SPECIFIER Code: AA 5000

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5005--

Condition Identifier: AAE\$INVALID_LOCK_USER_IDENT Code: AA 5005

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5010--

Condition Identifier: AAE\$FOUND_LOCK_NO_WAIT Code: AA 5010

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5015--

Condition Identifier: AAE\$ENTITY_WAS_ALREADY_LOCKED Code: AA 5015

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5020--

Condition Identifier: AAE\$DEADLOCK Code: AA 5020

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5025--

Condition Identifier: AAE\$SELF_DEADLOCK Code: AA 5025

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5030--

Condition Identifier: AAE\$NOT_PREVIOUSLY_LOCKED Code: AA 5030

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5035--

Condition Identifier: AAE\$EXPIRED_LOCK_EXISTS Code: AA 5035

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5040--

Condition Identifier: AAES\$NON_LAST_EXPIRED_UNLOCKED Code: AA 5040

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5045--

Condition Identifier: AAES\$LAST_EXPIRED_UNLOCKED Code: AA 5045

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5055--

Condition Identifier: AAES\$TOO_MANY_LOCKS_ALREADY Code: AA 5055

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5060--

Condition Identifier: AAES\$DEADLOCK_SYSTEM_ERROR Code: AA 5060

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--INFORMATIVE AA 5065--

Condition Identifier: AAES\$RESOLVE_DEPENDENCY_TIMEOUT Code: AA 5065

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--FATAL AA 6000--

Condition Identifier: AAES\$MIN_ANOMALOUS_CONDITION Code: AA 6000

Description: This is a system internal error.

User Action: Report this to your systems analyst.

--FATAL AA 6001-- Either the SCL variable AAV\$RESOLVE_TIME_LIMIT is defined, but is not an integer greater than 0, or CLP\$READ_VARIABLE returned an unexpected error status.

Condition Identifier: AAES\$BAD_RESOLVE_TIME_LIMIT Code: AA 6001

Description: If an SCL integer variable named AAV\$RESOLVE_TIME_LIMIT exists, it must contain the number of seconds that a lock request should wait before timing out.

User Action: Either delete the variable and use the default time limit (60 seconds) or assign a positive integer value (at least 1) to the variable.

Further Information: See the lock call descriptions in the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual. To read about SCL variables, see the NOS/VE System Usage manual.

--FATAL AA 6005-- AAM required space on the system heap in ring 4, but none was available.

Condition Identifier: AAES\$RING_4_SYSTEM_HEAP_FULL Code: AA 6005

Description: Internal error; internal keyed-file processing routines could not acquire space on the system heap as needed.

User Action: Report the software error.

--FATAL AA 6010-- The file {file} could not be opened because the lock manager has allocated more than half the space allowed by the FILE_LIMIT of the lock file. CLOSE is the only operation allowed on any file as long as this is true.

Condition Identifier: AAE\$LOCK_FILE_CROWDED_IN_OPEN Code: AA 6010

Description: The lock manager can process no more locks for this file.

User Action: Close the file, releasing all locks held by the instance of open. If possible, change the program so it minimizes the number of locks required and releases locks as soon as possible. If the program requires several concurrent key value locks, it should request a single file lock instead of individual key value locks. If necessary, attach the file for exclusive access (share_modes=none) so that no locks are necessary.

Further Information: See the lock processing description in the keyed-file interface section of the FORTRAN Usage or CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL AB 2-- (AB 0002) INPUT_OBJECT_TEXT file is \$NULL or nonexistent, no afterburning performed.

Condition Identifier: ABE\$IOT_FILE_NULL Code: AB 2

Description: The specification for the INPUT_OBJECT_TEXT parameter on the AFTERBURN_OBJECT_TEXT command must be an existing file or library containing the object code or object library to be afterburned.

User Action: Check that you have supplied a valid, existing, file or library name for the INPUT_OBJECT_TEXT parameter. The INPUT_OBJECT_TEXT parameter is positionally the first parameter on the AFTERBURN_OBJECT_TEXT command.

Further Information: The AFTERBURN_OBJECT_TEXT command is described in the NOS/VE Object Code Management manual.

--FATAL AB 3-- (AB 0003) The FILE_CONTENTS attribute for the INPUT_OBJECT_TEXT file must be OBJECT.

Condition Identifier: ABE\$IOT_FILE_CONTENTS Code: AB 3

Description: The INPUT_OBJECT_TEXT parameter must specify a file or library with a FILE_CONTENTS attribute of OBJECT.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify FILE_CONTENTS=OBJECT.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 5-- (AB 0005) The FILE_STRUCTURE attribute for the INPUT_OBJECT_TEXT file must be DATA or LIBRARY.

Condition Identifier: ABE\$IOT_FILE_STRUCTURE Code: AB 5

Description: The INPUT_OBJECT_TEXT parameter must specify a file or library file with a FILE_STRUCTURE attribute of DATA (for object code files) or LIBRARY (for object libraries).

User Action: You can use the SET_FILE_ATTRIBUTE or CHANGE_FILE_ATTRIBUTE commands to specify FILE_STRUCTURE=DATA (for an object file) or FILE_STRUCTURE=LIBRARY (for an object library).

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 6-- (AB 0006) The GLOBAL_ACCESS_MODE attribute for the INPUT_OBJECT_TEXT file must contain READ.

Condition Identifier: ABE\$IOT_GLOBAL_ACCESS_MODE Code: AB 6

Description: The INPUT_OBJECT_TEXT parameter must specify a file or library with GLOBAL_ACCESS_MODE that includes READ.

User Action: You can use the DISPLAY_FILE_ATTRIBUTE command to view the existing GLOBAL_ACCESS_MODE file attribute.

Further Information: Access modes are described in the NOS/VE System Usage manual.

--FATAL AB 7-- (AB 0007) The FILE_CONTENTS attribute for the OUTPUT_OBJECT_TEXT file must be either UNKNOWN or OBJECT.

Condition Identifier: ABE\$OOT_FILE_CONTENTS Code: AB 7

Description: The OUTPUT_OBJECT_TEXT parameter must specify a file or library with a FILE_CONTENTS attribute of UNKNOWN or OBJECT.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify FILE_CONTENTS=UNKNOWN or OBJECT.

--FATAL AB 8-- (AB 0008) The FILE_STRUCTURE attribute for the OUTPUT_OBJECT_TEXT file must be UNKNOWN or DATA.

Condition Identifier: ABE\$OOT_FILE_STRUCT_DATA Code: AB 8

Description: The OUTPUT_OBJECT_TEXT parameter must specify a file or library with a FILE_STRUCTURE attribute of UNKNOWN or DATA.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify FILE_STRUCTURE=UNKNOWN or FILE_STRUCTURE=OBJECT.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 9-- (AB 0009) The FILE_STRUCTURE attribute for the OUTPUT_OBJECT_TEXT file must be UNKNOWN or LIBRARY.

Condition Identifier: ABE\$OOT_FILE_STRUCT_LIBRARY Code: AB 9

Description: The OUTPUT_OBJECT_TEXT parameter must specify a file or library with a FILE_STRUCTURE attribute of UNKNOWN or LIBRARY.

User Action: You can use the SET_FILE_ATTRIBUTE or CHANGE_FILE_ATTRIBUTE command to specify FILE_STRUCTURE=UNKNOWN or FILE_STRUCTURE=LIBRARY.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System manual.

--FATAL AB 10-- (AB 0010) The GLOBAL_ACCESS_MODE attribute for the OUTPUT_OBJECT_TEXT file must contain MODIFY.

Condition Identifier: ABE\$OOT_GLOBAL_ACCESS_MODE Code: AB 10

Description: The OUTPUT_OBJECT_TEXT parameter must specify a file or library with a GLOBAL_ACCESS_MODE attribute that includes MODIFY.

User Action: You can use the DISPLAY_FILE_ATTRIBUTE command to view the existing GLOBAL_ACCESS_MODE file attributes.

Further Information: Access modes are described in the NOS/VE System Usage manual.

--FATAL AB 11-- (AB 0011) The FILE_CONTENTS attribute for the LIST file must be either UNKNOWN or LEGIBLE or LIST.

Condition Identifier: ABE\$LIST_FILE_CONTENTS Code: AB 11

Description: The LIST parameter must specify a file with a FILE_CONTENTS attribute of UNKNOWN, LEGIBLE, or LIST.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify

UNKNOWN
 LEGIBLE
 LIST

Further Information: the in the NOS/VE System Usage manual.FILE_ATTRIBUTE commands are described

--FATAL AB 12-- (AB 0012) The FILE_ORGANIZATION attribute for the LIST file must be SEQUENTIAL.

Condition Identifier: ABE\$LIST_FILE_ORGANIZATION Code: AB 12

Description: The LIST parameter must specify a file with a FILE_ORGANIZATION attribute of SEQUENTIAL.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify FILE_ORGANIZATION=SEQUENTIAL.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 13-- (AB 0013) The FILE_STRUCTURE attribute for the LIST file must be UNKNOWN or DATA.

Condition Identifier: ABE\$LIST_FILE_STRUCTURE Code: AB 13

Description: The LIST parameter must specify a file with a FILE_STRUCTURE attribute of UNKNOWN or DATA.

User Action: You can use the SET_FILE_ATTRIBUTE or CHANGE_FILE_ATTRIBUTE commands to specify FILE_STRUCTURE=UNKNOWN or FILE_STRUCTURE=DATA.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 14-- (AB 0014) The GLOBAL_ACCESS_MODE attribute for the LIST file must contain MODIFY.

Condition Identifier: ABE\$LIST_GLOBAL_ACCESS_MODE Code: AB 14

Description: The LIST parameter must specify a file with a GLOBAL_ACCESS_MODE attribute that includes MODIFY.

User Action: You can use the DISPLAY_FILE_ATTRIBUTE command to view the existing GLOBAL_ACCESS_MODE file attribute.

Further Information: Access modes are described in the NOS/VE System Usage manual.

--FATAL AB 19-- (AB 0019) The FILE_CONTENTS attribute for the {text} file number {text} must be either UNKNOWN or LEGIBLE or LIST.

Condition Identifier: ABE\$EP_FILE_CONTENTS Code: AB 19

Description: The file specified must have a FILE_CONTENTS attribute of UNKNOWN, LEGIBLE, or LIST.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify

FILE_CONTENTS=UNKNOWN

FILE_CONTENTS=LEGIBLE

FILE_CONTENTS=LIST

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 20-- (AB 0020) The FILE_ORGANIZATION attribute for the {text} file number {text} must be SEQUENTIAL.

Condition Identifier: ABE\$EP_FILE_ORGANIZATION Code: AB 20

Description: The file specified must have a FILE_ORGANIZATION attribute of SEQUENTIAL.

User Action: You can use the SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE command to specify FILE_ORGANIZATION=SEQUENTIAL.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the NOS/VE System Usage manual.

--FATAL AB 21-- (AB 0021) The FILE_STRUCTURE attribute for the {text} file number {text} must be UNKNOWN or DATA.

Condition Identifier: ABE\$EP_FILE_STRUCTURE Code: AB 21

Description: The file specified must have a FILE_STRUCTURE attribute of UNKNOWN or DATA.

User Action: You can use the CET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands to specify FILE_STRUCTURE=UNKNOWN or FILE_STRUCTURE=DATA.

Further Information: The SET_FILE_ATTRIBUTE and CHANGE_FILE_ATTRIBUTE commands are described in the SCL System Usage manual.

--FATAL AB 22-- (AB 0022) The GLOBAL_ACCESS_MODE attribute for the {text} file number {text} must contain READ.

Condition Identifier: ABE\$EP_GLOBAL_ACCESS_MODE Code: AB 22

Description: The specified file must contain a GLOBAL_ACCESS_MODE file attribute that includes READ.

User Action: You can use the DISPLAY_FILE_ATTRIBUTE command to view the GLOBAL_ACCESS_MODE attribute.

Further Information: The DISPLAY_FILE_ATTRIBUTE command is described in the NOS/VE System Usage manual.

--FATAL AB 23-- (AB 0023) INPUT_OBJECT_TEXT file is incomplete.

Condition Identifier: ABE\$INCOMPLETE_INPUT_FILE Code: AB 23

Description: The specified file is not complete or does not contain object code or an object library.

User Action: You can use the DISPLAY_FILE_ATTRIBUTE command to view the file attributes.

Further Information: The DISPLAY_FILE_ATTRIBUTE command is described in the NOS/VE System Usage manual.

--FATAL AB 301-- (AB 0301) The INPUT_OBJECT_TEXT file contains an obsolete version object library header.

Condition Identifier: ABE\$VERSION_MISMATCH Code: AB 301

Description: The INPUT_OBJECT_TEXT parameter must specify a file that does not contain an obsolete version header.

User Action:

--FATAL AB 304-- (AB 0304) Object library INPUT_OBJECT_TEXT has no section maps.

Condition Identifier: ABE\$MISSING_SECTION_MAPS Code: AB 304

Description: The INPUT_OBJECT_TEXT parameter must specify a library that includes section maps.

User Action: You can specify INCLUDE_BINARY_SECTION_MAPS=TRUE on the CREATE_MODULE command when creating a library.

Further Information: The CREATE_MODULE command is described in the NOS/VE System Usage manual.

--ERROR AM 0-- File {file} : {text} failed because the user/task has insufficient ring privilege.

Condition Identifier: AME\$RING_VALIDATION_ERROR Code: AM 0

Description: The ring attribute values for the task do not allow it to perform the requested operation on the file.

User Action: If possible, either change the appropriate ring_attributes value for the file to be accessed or the ring_attributes of the file containing the program executed to access the file.

--ERROR AM 5-- File {file} : {text} requires open with {text} access.

Condition Identifier: AME\$IMPROPER_ACCESS_ATTEMPT Code: AM 5

Description: The access mode of the file does not allow performance of the requested operation.

User Action: Correct the access mode value.

--ERROR AM 6-- File {file} : Null set {text} value of {text} improper.

Condition Identifier: AME\$NULL_SET_SPECIFIED Code: AM 6

Description: To validate the user, the call must specify the access modes to be granted the user.

User Action: Correct the access mode value.

--ERROR AM 7-- File {file} : Attempt to access segment with access not granted - {text}.

Condition Identifier: AME\$IMPROPER_SEGMENT_ACCESS Code: AM 7

Description: A segment access file was referenced improperly.

User Action: Correct the reference.

--ERROR AM 35-- File {file} : FILE BYTE ADDRESS parameter of {text} exceeded 2,147,483,647 (231 - 1).**

Condition Identifier: AME\$IMPROPER_FILE_BYTE_ADDRESS Code: AM 35

Description: The specified byte address exceeded the maximum file byte address.

User Action: Correct the specified parameter value.

--ERROR AM 65-- File {file} : {text} parameter of {text} had improper KEY(S) in array element(s): {text}.

Condition Identifier: AME\$IMPROPER_FILE_ATTRIB_KEY Code: AM 65

Description: Each key field must contain a valid file attribute identifier. The valid attribute identifiers begin with the characters AMC\$ followed by the attribute name.

User Action: Correct the value assigned to the key field.

--ERROR AM 70-- File {file} : {text} parameter of {text} had improper value(s) for the following: {text}.

Condition Identifier: AME\$IMPROPER_FILE_ATTRIB_VALUE Code: AM 70

Description: Each attribute field must contain a value valid for that attribute. Refer to the attribute description to determine the valid values.

User Action: Correct the value assigned to the attribute value field.

--ERROR AM 75-- File {file} : {text} may only be issued during open of a new file.

Condition Identifier: AME\$NOT_OPEN_NEW Code: AM 75

Description: The requested operation can only be issued when a file is being opened for the first time.

User Action: Check whether the file has been previously opened before attempting to set the attribute value(s) with this request.

--ERROR AM 80-- File {file} : FILE ATTRIBUTES parameter of {text} attempted to redefine value of following attribute(s): {text}.

Condition Identifier: AME\$ATTRIB_ALREADY_DEFINED Code: AM 80

Description: Using AMP\$ADD_TO_FILE_DESCRIPTION, a FAP can only define values for attributes whose values have not been explicitly defined for the file (attributes defined by default values or left undefined).

User Action: Before setting the attribute values, fetch the values and determine by the attribute source returned whether the FAP can define the attribute.

--ERROR AM 81-- {text}: File {file} must be opened previous to this request.

Condition Identifier: AME\$NOT_OLD_FILE Code: AM 81

Description: To change preserved file attributes, the file must have been previously opened.

User Action: If the file is a new file, use the appropriate call or command to set file attributes.

--ERROR AM 82-- {text}: File {file} must be attached with SHARE value of none.

Condition Identifier: AME\$IMPROPER_SHARE_SELECTION Code: AM 82

Description: When changing file attribute values, the file must be attached so that it cannot be attached at the same time by another user.

User Action: Change the attach request so that it specifies no share modes.

--ERROR AM 83-- {text}: New value of {text} must exceed preserved value for file {file}.

Condition Identifier: AME\$IMPROPER_NEW_ATTRIB_VALUE Code: AM 83

Description: The new file limit or record limit value must exceed the preserved value for the attribute.

User Action: Increase the specified attribute value.

--ERROR AM 84-- File {file} : The following file attributes cannot be overridden using {text}: {text}.

Condition Identifier: AME\$IMPROPER_OVERRIDE_ATTEMPT Code: AM 84

Description: The only preserved file attributes that can be overridden are block type, record type, file organization and ring attributes.

User Action: Do not try to override the file attributes listed in the message.

--ERROR AM 85-- Missing or damaged file attributes.

Condition Identifier: AME\$DAMAGED_FILE_ATTRIBUTES Code: AM 85

Description: The file label contains invalid information.

User Action: Delete the file and try to reload it, if a backup file exists.

--ERROR AM 86-- File {file} : Level {text} file attributes not {text} compatible.

Condition Identifier: AME\$INCOMPATIBLE_ATTRIBUTES Code: AM 86

Description: The file cannot be attached or restored because its file label is incompatible with the revision level of the file labels on the current system.

User Action: Delete the file.

--ERROR AM 95-- File {file} : Structural attribute values specified by {text} do not match the preserved attribute values for the following: {text}.

Condition Identifier: AME\$ATTRIBUTE_VALIDATION_ERROR Code: AM 95

Description: The attempt to open the file determined that one or more preserved file attributes do not have the values specified on the open call.

User Action: Ensure that the correct file is specified.

--ERROR AM 96-- Cannot connect {file} and {file} because of incompatible attributes: {text}.

Condition Identifier: AME\$INCOMPATIBLE_FILE_CONNECT Code: AM 96

Description: The file content and file structure of the files to be connected must either be identical or one of the two files must have unknown file content or file structure. The access mode selection sets of both files must also intersect.

User Action: Change the attributes of the files so they are compatible with each other.

--ERROR AM 100-- File {file} : Limit of {text} local files exceeded.

Condition Identifier: AME\$LOCAL_FILE_LIMIT Code: AM 100

Description: You attempted to define another local file beyond your local file limit.

User Action: Return one or more local files.

--ERROR AM 101-- File {file} : Limit of {text} concurrent instances of open of files in the same task was exceeded. This limit is enforced by {text} Management.

Condition Identifier: AME\$CONCURRENT_OPEN_LIMIT Code: AM 101

Description: A task allows only 200 concurrent instances of open of files.

User Action: Close some instances of open.

--ERROR AM 105-- File {file} : ACCESS LEVEL parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_ACCESS_LEVEL Code: AM 105

Description: The only valid access levels are segment access and record access.

User Action: Correct the access level specification.

--ERROR AM 110-- File {file} : Maximum block length must equal or exceed minimum block length with user specified blocking.

Condition Identifier: AME\$MBL_LESS_THAN_MIBL Code: AM 110

Description: The block_type attribute value is user-specified, but the max_block_length attribute value is less than the min_block_length attribute value.

User Action: Either increase the max_block_length attribute value or decrease the min_block_length attribute value.

--ERROR AM 115-- File {file} : Maximum block length must equal or exceed the ANSI fixed (F) record length with user specified blocking.

Condition Identifier: AME\$MBL_LESS_THAN_MRL Code: AM 115

Description: The record_type attribute value is F record type, but the max_block_length attribute value is less than the max_record_length attribute value (the fixed record length).

User Action: Either increase the max_block_length attribute value or decrease the max_record_length attribute value.

--ERROR AM 120-- File {file} : Access mode of {text} conflicts with file access permissions.

Condition Identifier: AME\$NO_PERMISSION_FOR_ACCESS Code: AM 120

Description: The access mode was not granted when the file was attached.

User Action: Request the access mode when the file is attached.

--ERROR AM 121-- File {file} is already open for {text} access which conflicts with the specified {text} access mode.

Condition Identifier: AME\$CONFLICTING_FILE_ACCESS Code: AM 121

Description: The access mode specified on the open call does not match the access for which the file is currently opened.

User Action: Open the file with the access mode that agrees with its current access permissions.

--ERROR AM 130-- File {file} : An undefined (U) or ANSI fixed (F) record type cannot be overridden by variable (V) record type.

Condition Identifier: AME\$IMPROPER_RECORD_OVERRIDE Code: AM 130

Description: Variable (V) record type cannot be used to read another record type.

User Action: Determine and specify the correct record type for the file.

--ERROR AM 135-- File {file} : Overriding block type, file organization or record type is improper with write access.

Condition Identifier: AME\$IMPROPER_OVERRIDE_ACCESS Code: AM 135

Description: The specified record type or block type attribute value differed from the value preserved with the file and the access mode requested was not read only.

User Action: Either change the access mode set requested to read access only or do not attempt to change the record type or block type attributes.

--ERROR AM 140-- File {file} : Conditions under which file organization can be overridden with write access were not met.

Condition Identifier: AME\$IMPROPER_WRITE_OVERRIDE Code: AM 140

Description: With write access, sequential file organization is the only value that can be overridden.

User Action: Do not attempt to change the file organization attribute.

--ERROR AM 141-- File {file} : Overriding user specified blocking with system specified blocking is improper with variable (V) or ANSFI fixed (F) record type.

Condition Identifier: AME\$IMPROPER_US_BLOCK_OVERRIDE Code: AM 141

Description: A block_type attribute value of system-specified blocking cannot be specified when the preserved block_type attribute value is user-specified blocking and the record_type attribute is not U records.

User Action: Remove the block_type attribute value or specify the record_type attribute value as the U record type.

--ERROR AM 142-- File {file} : Overriding system specified blocking with user specified blocking is improper.

Condition Identifier: AME\$IMPROPER_SS_BLOCK_OVERRIDE Code: AM 142

Description: A block_type attribute value of user-specified blocking cannot be specified when the preserved block_type attribute value is system-specified blocking.

User Action: Remove the block_type attribute value.

--ERROR AM 143-- File {file} : Sequential and byte addressable file organizations cannot be overridden by indexed sequential file organization.

Condition Identifier: AME\$IMPROPER_FO_OVERRIDE Code: AM 143

Description: A file_organization attribute value of byte addressable can be changed to sequential and vice versa.

User Action: To store the file data on an indexed sequential file, you must write the file records on a new indexed sequential file.

--ERROR AM 150-- File {file} has not been previously opened. The initial open request for a file must permit file creation. The initial attachment of a mass storage file that has never been opened must also include append in the access modes.

Condition Identifier: AME\$NEW_FILE_REQUIRES_APPEND Code: AM 150

Description: A new file is empty; therefore, the system assumes the task opens the file in order to write data at its end of information which requires append access.

User Action: Request append access on the open call.

--ERROR AM 155-- File {file} : \$EOI open position required for append-only access.

Condition Identifier: AME\$IMPROPER_APPEND_OPEN Code: AM 155

Description: Because the file is opened with only append access, the only valid file position is at its end of information.

User Action: Change the open_position file attribute value to end of information.

--ERROR AM 165-- File {file} : Segment access not allowed with indexed sequential file organization.

Condition Identifier: AME\$FO_ACCESS_LEVEL_CONFLICT Code: AM 165

Description: Only record level access is valid for indexed sequential files.

User Action: Change the access level parameter to specify record access.

--ERROR AM 175-- File {file} : {text} file organization not supported on {text} device class.

Condition Identifier: AME\$FO_DEVICE_CLASS_CONFLICT Code: AM 175

Description: Only sequential file organization is valid for files assigned to the magnetic tape or interactive terminal device classes.

User Action: Either change the device class assignment or change the file_organization attribute value for the file.

--ERROR AM 178-- File {file} : {text} file_organization is not supported for {text} records.

Condition Identifier: AME\$FO_RECORD_TYPE_CONFLICT Code: AM 178

Description: Only sequential file organization is valid for files whose record type is trailing character delimited.

User Action: Change the file organization or record type attribute value for the file.

--ERROR AM 180-- File {file} : Segment access not allowed on {text} device class.

Condition Identifier: AME\$NOT_VIRTUAL_MEMORY_DEVICE Code: AM 180

Description: To use segment access, the file must not be assigned to the tape or terminal device classes.

User Action: Either specify a mass storage file or change the requested access level to record access.

--ERROR AM 195-- File {file} : Unable to load file access procedure {text}.

Condition Identifier: AME\$UNABLE_TO_LOAD_FAP Code: AM 195

Description: The loader could not find the specified error exit procedure.

User Action: Ensure that the specified error exit procedure exists within the task object library list in uppercase letters or as an entry point in the task itself.

--ERROR AM 197-- File {file} : A required module for file organization {text} could not be loaded because object library \$LOCAL.AAF\$44D_LIBRARY is not in the program library list. Add object library using SET_PROGRAM_ATTRIBUTES.

Condition Identifier: AME\$KEYED_FILE_FAP_MISSING Code: AM 197

Description: The required module for the requested Advanced_Access_Method file organization could not be loaded because object library \$LOCAL.AAF\$44D_LIBRARY is not in the program library list.

User Action: Add \$LOCAL.AAF\$44D_LIBRARY to the program library list using SET_PROGRAM_ATTRIBUTES.

--ERROR AM 200-- File {file} : Unable to load collate table named {text}.

Condition Identifier: AME\$UNABLE_TO_LOAD_COLLATE_TABL Code: AM 200

Description: The loader could not find the specified collation table.

User Action: Ensure that the collate_table attribute value specifies the collation table name in uppercase letters and ensure that the specified collation table exists within the task object library list or as an entry point in the task itself.

--ERROR AM 205-- File {file} : Unable to load error exit procedure named {text}.

Condition Identifier: AME\$UNABLE_TO_LOAD_ERROR_EXIT Code: AM 205

Description: The loader could not find the specified error exit procedure.

User Action: Ensure that the error_exit_name attribute value specifies the error exit procedure name in uppercase letters and ensure that the specified error exit procedure exists within the task object library list or as an entry point in the task itself.

--ERROR AM 210-- File {file} : Unable to load label exit procedure named {text}.

Condition Identifier: AM\$UNABLE_TO_LOAD_LABEL_EXIT Code: AM 210

Description: The loader could not find the specified label exit procedure.

User Action: Ensure that the label exit procedure exists within the task object library list or as an entry point in the task itself. Ensure that the label_exit_name attribute value specifies the label exit procedure name in uppercase letters.

--ERROR AM 215-- File {file} : Limit of {text} tasks sharing an interactive file exceeded.

Condition Identifier: AM\$TERMINAL_TASK_LIMIT Code: AM 215

Description: The task cannot open the file until another task closes the file.

User Action: If appropriate, define another terminal file for use by the task.

--ERROR AM 220-- File {file} : Limit of {text} concurrently used {text} track, {text} cpi density tapes exceeded.

Condition Identifier: AM\$CONCURRENT_TAPE_LIMIT Code: AM 220

Description: The task attempted to exceed the job reservation limit for tape units.

User Action: Enter an SCL RESERVE_RESOURCE command to reserve the appropriate number of tape units for the task.

--ERROR AM 221-- File {file} : Multiple instances of open of tape file are improper.

Condition Identifier: AM\$MULTIPLE_OPEN_OF_TAPE Code: AM 221

Description: Only one instance of open of a tape file is valid at a time.

User Action: Close the previous instance of open before opening the tape file again.

--ERROR AM 222-- File {file} : An open for write access to a tape file requires a write-ring in the reel but none was requested.

Condition Identifier: AM\$NO_WRITE_RING Code: AM 222

Description: A write ring must be inserted in the tape reel in order to open the file for write access.

User Action: Insert a write ring in the tape reel.

--ERROR AM 225-- File {file} : Null set is improper access_mode value.

Condition Identifier: AM\$NULL_ACCESS_MODE Code: AM 225

Description: To perform file operations, at least one access mode must be requested.

User Action: Correct the access mode value.

--ERROR AM 230-- File {file} : {text} blocking is not supported for {text} records.

Condition Identifier: AM\$RT_BLOCK_TYPE_CONFLICT Code: AM 230

Description: A file with a record type of trailing character delimited is only valid with system specified blocking.

User Action: Change the block type or record type of the file.

--ERROR AM 235-- File {file} : {text} device_class is not supported for {text} records.

Condition Identifier: AM\$RT_DEVICE_CLASS_CONFLICT Code: AM 235

Description: A file with record type of trailing character delimited can only be assigned to a mass storage device.

User Action: Change either the record type attribute value of the file or the device class assignment.

--ERROR AM 240-- File {file} : {text} blocking and {text} records is not supported for MAGNETIC TAPE DEVICE.

Condition Identifier: AME\$UNSUPPORTED_BT_AND_RT Code: AM 240

Description: The specified combination of block type and record type are not supported for ANSI labelled tape files.

User Action: Specify a different block type and record type combination.

--ERROR AM 395-- File {file} : {text} improper with {text} device class.

Condition Identifier: AME\$IMPROPER_DEVICE_CLASS Code: AM 395

Description: For the requested operation, the file must be assigned to the magnetic tape device class.

User Action: Check that the correct file is specified and that it has been assigned to the tape device class.

--ERROR AM 405-- {text} issued but file_identifier parameter value is improper.

Condition Identifier: AME\$IMPROPER_FILE_ID Code: AM 405

Description: The system did not recognize the file identifier as that of an open file.

User Action: Specify the file identifier returned by the open call. Do not change the file identifier value.

--ERROR AM 415-- File {file} : Working_storage_length parameter of {text} had improper value specified.

Condition Identifier: AME\$IMPROPER_WSL_VALUE Code: AM 415

Description: The value of the working_storage_length parameter is improper.

User Action: Provide a proper working_storage_length value.

--ERROR AM 425-- File {file} : ACCESS_INFORMATION parameter of {text} had improper KEY(S) in array element(s): {text}.

Condition Identifier: AME\$IMPROPER_ACCESS_INFO_KEY Code: AM 425

Description: A key field in the access information array contained an unknown identifier.

User Action: Correct the key field so that it specifies a valid access information item identifier.

--ERROR AM 430-- File {file} : {text} operation not supported for {text} record type and {text} block type.

Condition Identifier: AME\$UNSUPPORTED_OPERATION Code: AM 430

Description: The operation requested is not supported by the record type and block type combination being used.

User Action: Use a different request that supports the block and record type, or block type to one that supports the request.

--ERROR AM 433-- File {file} : {text} block type/{text} record type no longer supported. Use FAP=AMP\$US_BLK_VAR_READ_ONLY_FAP to read file to another block/record type.

Condition Identifier: AME\$DELETED_BT_RT Code: AM 433

Description: the combination of user-specified block type and variable record type is no longer supported.

User Action: Use FAP=AMP\$US_BLK_VAR_READ_ONLY_FAP to read the file to another block type/record type.

--ERROR AM 435-- File {file} : {text} not implemented{text}.

Condition Identifier: AME\$UNIMPLEMENTED_REQUEST Code: AM 435

Description: The specified request is not implemented in this version of the system.

User Action: Check that the correct request name is specified.

--ERROR AM 445-- File {file} : {text} issued but opened for {text} access.

Condition Identifier: AME\$CONFLICTING_ACCESS_LEVEL Code: AM 445

Description: The request failed because the file was not opened for the appropriate access mode.

User Action: Change the access mode set requested on the open call so that the set includes all access modes required for the instance of open.

--ERROR AM 455-- File {file} : {text} issued but not opened for input.

Condition Identifier: AME\$IMPROPER_INPUT_ATTEMPT Code: AM 455

Description: The file was not opened with read access.

User Action: Change the open call so that it requests read access.

--ERROR AM 460-- File {file} : {text} attempted after output on {text} file.

Condition Identifier: AME\$INPUT_AFTER_OUTPUT Code: AM 460

Description: An attempt to read from the file was made after writing to the file.

User Action: Change the program to reposition the file prior to an attempt to read.

--ERROR AM 465-- File {file} : SKIP OPTION parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SKIP_OPTION Code: AM 465

Description: The valid skip options are skip to end of record and no skip.

User Action: Correct the skip option value.

--ERROR AM 470-- File {file} : BYTE ADDRESS parameter of {text} not start of {text} record.

Condition Identifier: AME\$IMPROPER_RECORD_ADDRESS Code: AM 470

Description: To read a record directly, the byte address specified must be that of the beginning of a record.

User Action: Determine the correct address of the record. If the record addresses are unknown, the records must be read sequentially to determine their addresses and a directory created for subsequent direct access.

--ERROR AM 475-- File {file} : {text} issued but current byte address is not start of a {text} record header.

Condition Identifier: AME\$IMPROPER_RECORD_HEADER Code: AM 475

Description: The file is not positioned at a record_header for a VARIABLE record_type file. The file is either mispositioned, damaged, or the file is not VARIABLE record_type.

User Action: Make sure the file being read is record_type VARIABLE. This is most likely to happen when reading tape files.

Make sure that the position is correct for reading a variable record. It is most likely that a file could be mispositioned if not read using file_organization = SEQUENTIAL.

Verify data in the file. Check to see if there are any gaps in the data or if the file has been truncated or extended. A file can sometimes be truncated or eoi set beyond the expected eoi if the the system has crashed.

--ERROR AM 555-- File {file} : {text} issued but not opened for output.

Condition Identifier: AME\$IMPROPER_OUTPUT_ATTEMPT Code: AM 555

Description: The file was not opened with the appropriate append or modify access.

User Action: Specify the appropriate access modes on the open call.

--ERROR AM 560-- File {file} : Length of {text} record exceeds maximum block length.

Condition Identifier: AME\$RECORD_EXCEEDS_MBL Code: AM 560

Description: The task attempted to write a record longer than the maximum block length for the file.

User Action: If copying a file, increase the max_block_length attribute value for the output file. If writing records to a file, either increase the max_block_length value or, if possible, decrease the working storage length for the record.

--ERROR AM 565-- File {file} : TERM OPTION parameter of {text} was improper.

Condition Identifier: AME\$IMPROPER_TERM_OPTION Code: AM 565

Description: The valid termination options are AMC\$START, AMC\$CONTINUE, and AMC\$TERMINATE.

User Action: Correct the term_option value.

--ERROR AM 570-- File {file} : TERM OPTION OF {text} was CONTINUE but file position was not mid-record.

Condition Identifier: AME\$IMPROPER_CONTINUE Code: AM 570

Description: To write the middle part of a record, the file position must be within a record.

User Action: Either change the term_option parameter value or ensure that the first part of the record is written before this call.

--ERROR AM 575-- File {file} : A user-specified blocked file must be created and appended sequentially.

Condition Identifier: AME\$FO_BLOCK_TYPE_CONFLICT Code: AM 575

Description: A file whose block_type attribute value indicates user-specified blocking must be opened for record access and its records written sequentially with AMP\$PUT_NEXT and AMP\$PUT_PARTIAL calls.

User Action: Change the access level to record access and use only sequential access calls to write the file.

--ERROR AM 580-- File {file} : {text} requires append permission to position beyond eoi.

Condition Identifier: AME\$POSITION_BEYOND_EOI Code: AM 580

Description: To position beyond EOI, the requested operation requires append permission for the file.

User Action: Specify append access mode on the open call.

--ERROR AM 585-- File {file} : Improper attempt to position beyond the file limit using {text}.

Condition Identifier: AME\$POSITION_BEYOND_FILE_LIMIT Code: AM 585

Description: Positioning beyond the file's preserved file limit is not allowed.

User Action: Either specify a greater file limit on the first open of the file or do not attempt to position beyond the file limit.

--ERROR AM 590-- File {file} : Improper attempt to write beyond the file limit using {text}.

Condition Identifier: AME\$PUT_BEYOND_FILE_LIMIT Code: AM 590

Description: Writing beyond the file limit preserved with the file is not allowed.

User Action: Specify a greater file limit on the first open of the file.

--ERROR AM 595-- File {file} : Length of {text} record must be identical to the length of the record being replaced by {text}.

Condition Identifier: AME\$RECORD_UNEQUAL_TO_PREVIOUS Code: AM 595

Description: The length of a record that is being written to replace a record must be equal in length to the record being replaced.

User Action: Specify an equal WORKING_STORAGE_LENGTH.

--ERROR AM 600-- File {file} : File not positioned at the end of a record prior to {text} request.

Condition Identifier: AME\$IMPROPER_FILE_POSITION Code: AM 600

Description: A file must be positioned at the end of a record prior to issuing a REPLACE_PREVIOUS_RECORD request.

User Action: Position file to END_OF_RECORD.

--ERROR AM 605-- File {file} : BYTE_ADDRESS parameter of {text} must be the address of a record boundary.

Condition Identifier: AME\$IMPROPER_SEEK_ADDRESS Code: AM 605

Description: The BYTE_ADDRESS parameter specified in a byte-addressable program interface (AMP\$SEEK_DIRECT, AMP\$GET_DIRECT, or AMP\$PUT_DIRECT) was not legal. The byte address must be a record boundary (i.e., BOI, EOR, or EOI).

User Action: Specify a byte address that is a record boundary. This is the BYTE_ADDRESS parameter specified on byte-addressable write requests (AMP\$PUT_DIRECT) or returned by sequential write requests (AMP\$PUT_NEXT or AMP\$PUT_PARTIAL_START).

--ERROR AM 655-- File {file} : UNIT parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SKIP_UNIT Code: AM 655

Description: The valid skip units are records, partitions, and tapemarks.

User Action: Correct the skip unit value.

--ERROR AM 660-- File {file} : DIRECTION parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SKIP_DIRECTION Code: AM 660

Description: The valid skip directions are forward and backward.

User Action: Correct the skip direction value.

--ERROR AM 665-- File {file} : COUNT parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SKIP_COUNT Code: AM 665

Description: The skip count value is not within the valid range.

User Action: Correct the skip count value.

--ERROR AM 670-- File {file} : {text} requires READ permission.

Condition Identifier: AME\$SKIP_REQUIRES_READ_PERM Code: AM 670

Description: To reposition the file, the task must open the file for read access.

User Action: Request read access on the open call.

--ERROR AM 675-- File {file} : SKIP {text} {text} for {text} records and {text} blocking not supported.

Condition Identifier: AME\$UNSUPPORTED_SKIP Code: AM 675

Description: Skipping by partitions is valid only for V record type. Skipping with system-specified blocking is valid only for V and F record types.

User Action: Eliminate the skip request or change the block type and record type of the file.

--ERROR AM 680-- File {file} : A SKIP cannot be performed on a file created with {text} file organization.

Condition Identifier: AME\$CONFLICTING_FO Code: AM 680

Description: A skip operation can only be performed on a file created with sequential file organization.

User Action: Do not attempt to reposition the file using skip operations.

--ERROR AM 755-- File {file} : {text} issued but {text} records do not support partitioning.

Condition Identifier: AME\$PARTITIONING_UNSUPPORTED Code: AM 755

Description: Only the variable (V) record type supports file partitioning.

User Action: Either change the record_type file attribute or do not attempt to write a partition delimiter on the file.

--ERROR AM 775-- File {file} : POINTER KIND parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_POINTER_KIND Code: AM 775

Description: The valid pointer kinds are pointers to a cell, a heap, or a segment.

User Action: Correct the specified pointer kind value.

--ERROR AM 780-- File {file} : SEGMENT POINTER parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SEGMENT_POINTER Code: AM 780

Description: To perform the requested operation, the segment pointer cannot be NIL.

User Action: Either request a different segment pointer or do not attempt to set the file position or end of information.

--ERROR AM 785-- File {file} : {text} attempted beyond end of information.

Condition Identifier: AME\$SET_POS_BEYOND_EOI Code: AM 785

Description: The current file position cannot be beyond its current end of information.

User Action: Either change the file position requested to a position before the end of information or extend the end of information before setting the file position.

--ERROR AM 790-- File {file} : {text} requires APPEND permission to increase the end of information byte address.

Condition Identifier: AME\$SET_EOI_NEEDS_APPEND Code: AM 790

Description: To extend the file, the task must open the file for append access.

User Action: Specify append access mode on the open call.

--ERROR AM 795-- File {file} : {text} requires SHORTEN permission to decrease the end of information byte address.

Condition Identifier: AME\$SET_EOI_NEEDS_SHORTEN Code: AM 795

Description: To shorten the file, the task must open the file for shorten access.

User Action: Specify append access mode on the open call.

--ERROR AM 800-- File {file} : {text} does not accept an adaptable heap pointer.

Condition Identifier: AME\$SET_ON_ADAPTABLE_HEAP Code: AM 800

Description: When the segment pointer is a heap pointer, the file has no file position or end of information recorded.

User Action: Either request a different segment pointer kind or do not attempt to set the file position or end of information.

--ERROR AM 801-- File {file} : {text} issued for empty, read-only segment.

Condition Identifier: AME\$READ_OF_EMPTY_SEGMENT Code: AM 801

Description: A segment can only be read when it is not empty.

User Action: Do not try to read an empty segment.

--ERROR AM 802-- File {file} : {text} issued but empty, writable segment not opened for append access mode.

Condition Identifier: AME\$WRITE_OF_EMPTY_SEGMENT Code: AM 802

Description: To perform the requested operation, the segment file to be written must be opened for append access.

User Action: Change the access mode set to include append.

--ERROR AM 810-- File {file} : Segment number within the SEGMENT POINTER parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_SEGMENT_NUMBER Code: AM 810

Description: The segment number field within the segment pointer has been changed.

User Action: Do not change the segment number field within the segment pointer returned by the AMP\$GET_SEGMENT_POINTER call.

--ERROR AM 875-- File {file} : {text} is not a {text} access request.

Condition Identifier: AME\$FILE_ORGANIZATION_CONFLICT Code: AM 875

Description: To perform the requested operation, the file_organization attribute value for the file must be byte addressable.

User Action: Check that the correct file identifier is specified and, if so, change the file_organization attribute to byte addressable.

--ERROR AM 877-- File {file} : {text} beyond EOI requires byte_addressable file_organization.

Condition Identifier: AME\$SEEK_BEYOND_EOI_FO_CONFLICT Code: AM 877

Description: The byte address specified on an AMP\$SEEK_DIRECT was beyond END_OF_INFORMATION and the FILE_ORGANIZATION was not BYTE_ADDRESSABLE.

User Action: FILE_ORGANIZATION must be BYTE_ADDRESSABLE or the byte address specified must be less than or equal to END_OF_INFORMATION.

--ERROR AM 906-- File {file} : Maximum cancellable input exceeded.

Condition Identifier: AME\$MAX_CANCELLABLE_INPUT Code: AM 906

Description: The working storage area has indeterminate data.

User Action: Re-issue the request.

--ERROR AM 915-- File {file} : CALL_BLOCK parameter from file access procedure had improper OPERATION specification.

Condition Identifier: AME\$IMPROPER_FAP_OPERATION Code: AM 915

Description: The operation field of the call_block variable does not contain a valid value.

User Action: If the call_block is passed to the FAP, do not change the operation value. If the call_block is generated within the FAP, ensure that a valid operation identifier is specified for the field.

--ERROR AM 920-- File {file} : Redundant {text} request issued by file access procedure.

Condition Identifier: AME\$REDUNDANT_STRUCTURE_POINTER Code: AM 920

Description: An AMP\$STORE_FAP_POINTER has already defined a data structure for the FAP.

User Action: Issue an AMP\$STORE_FAP_POINTER call only when the FAP is called and passed the file open call.

--ERROR AM 925-- File {file} : Layer number parameter of {text} improper.

Condition Identifier: AME\$IMPROPER_LAYER_NUMBER Code: AM 925

Description: The layer number specified is incorrect.

User Action: Do not change the layer number passed to the FAP. Specify the layer number passed to the FAP on the AMP\$ACCESS_METHOD call.

--ERROR AM 1015-- {text} was issued but {file} is open.

Condition Identifier: AME\$FILE_NOT_CLOSED Code: AM 1015

Description: The system could not perform the requested operation because all instances of open have not been closed for the file.

User Action: Close all instances of open for the file before attempting the operation.

--ERROR AM 1016-- {text} was issued for file, {file}, which does not exist.

Condition Identifier: AME\$FILE_NOT_KNOWN Code: AM 1016

Description: The system could not perform the operation because the specified local file name has not been defined within the job.

User Action: Check that the correct local file name is specified.

--ERROR AM 1017-- File {file} is already known within the job.

Condition Identifier: AME\$FILE_KNOWN Code: AM 1017

Description: The specified local_file_name must not be defined within the job.

User Action: Detach the local file.

--ERROR AM 1045-- File {file} : {text} attempted at or beyond end of information.

Condition Identifier: AME\$INPUT_AFTER_EOI Code: AM 1045

Description: The task attempted to read the file again after end-of-information was returned as the file position.

User Action: Check for the end-of-information file position and do not attempt to read the file after the EOI status is returned (without repositioning the file).

--ERROR AM 1050-- File {file} : {text} terminated due to unrecovered device error. {text}

Condition Identifier: AME\$UNRECOVERED_READ_ERROR Code: AM 1050

Description: An unrecoverable error occurred on a read from a tape file.

User Action: For specific information, refer to the Sequential and Byte-Addressable Files Usage manual under the topic Tape Failure Isolation Information.

--ERROR AM 1055-- File {file} : Data moved to working storage area taken from block subject to unrecovered device error. {text}

Condition Identifier: AME\$ACCEPT_BAD_BLOCK Code: AM 1055

Description: The data in the working storage area may not be correct. The system detected an unrecoverable error while reading the block.

User Action: Check the data in the working storage area for usefulness.

--ERROR AM 1060-- File {file} : {text} block on file exceeds max block length of {text}.

Condition Identifier: AME\$INPUT_BLOCK_EXCEEDS_MAXBL Code: AM 1060

Description: The length of a block on the tape exceeds the maximum block length of the tape file.

User Action: Increase the maximum block length to a value that is equal to or greater than the longest block on the tape.

--ERROR AM 1145-- File {file} : {text} terminated due to unrecovered device error. {text}

Condition Identifier: AME\$UNRECOVERED_WRITE_ERROR Code: AM 1145

Description: An unrecoverable error occurred on a write to a tape file.

User Action: For specific information, refer to the Sequential and Byte-Addressable Files Usage manual under the topic Tape Failure Isolation Information.

--ERROR AM 1245-- File {file} : SKIP backward {text} encountered beginning of information before COUNT exhausted.

Condition Identifier: AME\$SKIP_ENCOUNTERED_BOI Code: AM 1245

Description: The file could not be repositioned as requested because insufficient skip units existed between the former position and the beginning of information. The file is left positioned at its beginning of information.

User Action: Correct the skip count parameter value.

--ERROR AM 1250-- File {file} : SKIP backward records encountered beginning of partition before COUNT exhausted.

Condition Identifier: AME\$SKIP_ENCOUNTERED_BOP Code: AM 1250

Description: The file could not be repositioned as requested because insufficient records existed between the former position and the beginning of the partition. The file is left positioned at the beginning of the partition.

User Action: Correct the skip count parameter value.

--ERROR AM 1255-- File {file} : SKIP forward records encountered an end of partition before COUNT exhausted.

Condition Identifier: AME\$SKIP_ENCOUNTERED_EOP Code: AM 1255

Description: The file could not be repositioned as requested because insufficient records existed between the former position and the end of the partition. The file is left positioned at the end of the partition.

User Action: Correct the skip count value.

--ERROR AM 1260-- File {file} : SKIP forward {text} encountered end of information before COUNT exhausted.

Condition Identifier: AME\$SKIP_ENCOUNTERED_EOI Code: AM 1260

Description: The file could not be repositioned as requested because insufficient skip units existed between the former position and the end of information. The file is left positioned at its end of information.

User Action: Correct the skip count value.

--ERROR AM 1265-- File {file} : SKIP backward by tapemarks encountered beginning-of-volume.

Condition Identifier: AME\$SKIP_ENCOUNTERED_BOV Code: AM 1265

Description: A skip backward on a labelled or nonstandard labelled tape file could not be completed as requested because insufficient embedded tapemarks existed between the former position and the beginning of the volume. The tape is left positioned at the beginning of the volume.

User Action: Correct the skip count value.

--ERROR AM 1275-- File {file} : Tape volume position is uncertain after {text}. {text}

Condition Identifier: AME\$UNCERTAIN_TAPE_POSITION Code: AM 1275

Description: The system no longer knows the tape position. This could be due to a tape unit error.

User Action: For specified information, refer to the Sequential and Byte-Addressable Files Usage manual under the topic Tape Failure Isolation Information.

--ERROR AM 1451-- File {file} : The maximum block length for this request is larger than the system can handle at this page size.

Condition Identifier: AME\$CANNOT_LOCK_TAPE_PAGES Code: AM 1451

Description: An attempt was made to open a tape file with a maximum block length (MAXBL) larger than is supported for the page size for which the system is configured.

User Action: Ask site personnel to change the system configuration (larger page size) to support longer tape blocks.

--FATAL AM 1452-- File {file} : Internal tape block manager malfunction. {text}.

Condition Identifier: AME\$TAPE_BLOCK_MGR_MALFUNCTION Code: AM 1452

Description: An error internal to the tape system has been detected.

User Action: Write a PSR and include the appropriate supporting documentation.

--ERROR AM 1453-- File {file} : The maximum block length for this file is larger than the maximum size the tape driver can handle in this configuration.

Condition Identifier: AME\$TAPE_DRIVER_NOT_CAPABLE Code: AM 1453

Description: The maximum block length specified for the tape file exceeds the length supported by the tape subsystem in this configuration.

User Action: Ask site personnel to change the tape configuration to dual PP to support longer tape blocks.

--FATAL AM 1454-- File {file} : Internal tape record manager malfunction. {text}.

Condition Identifier: AME\$TAPE_RCD_MGR_MALFUNCTION Code: AM 1454

Description: An error internal to the tape system has been detected.

User Action: Write a PSR and include the appropriate supporting documentation.

--ERROR AM 1455-- File {file} : End of tape encountered - operation completed.

Condition Identifier: AME\$END_OF_TAPE_OP_COMPLETED Code: AM 1455

Description: A physical end of tape was encountered while writing to a tape file. The write operation completed normally and all of the tape blocks were written to the current volume.

User Action: Call AMP\$CLOSE_VOLUME to advance to the next tape volume and continue writing.

--ERROR AM 1456-- File {file} : End of tape encountered - operation inhibited.

Condition Identifier: AME\$END_OF_TAPE_OP_INHIBITED Code: AM 1456

Description: A physical end of tape was encountered while writing to a tape file. The write operation did not complete and one or more of the buffered tape blocks were not written to the current volume.

User Action: Call AMP\$CLOSE_VOLUME to advance to the next tape volume. Buffered tape blocks that were not written to on the current volume will be included on the next tape volume. The current write operation must be reissued after advancing to the next volume.

--ERROR AM 1457-- File {file} : {text} encountered end of volume list.

Condition Identifier: AME\$TAPE_END_OF_VOLUME_LIST Code: AM 1457

Description: An attempt has been made to advance beyond the end of the tape volume list.

User Action: Add one or more tape volumes to the tape volume list.

--ERROR AM 1458-- File {file} : {text} cannot be performed with a MAX_BLOCK_LENGTH within 20 pages of the job working set limit.

Condition Identifier: AME\$MAXBL_EXCEEDS_WS_LIMIT Code: AM 1458

Description: The maximum block length specified for a tape file is within 20 pages of the job's working set limit.

User Action: Ask site personnel to increase the user's working set limit.

--WARNING AM 1476-- File {file} : Parameter FILE_SEQUENCE_NUMBER is ignored for FILE_SET_POSITION = FILE_IDENTIFIER_POSITION.

Condition Identifier: AME\$FILE_SEQ_NUMBER_IGNORED Code: AM 1476

Description: A FILE_SEQUENCE_NUMBER parameter has been specified on the CHANGE_TAPE_LABEL_ATTRIBUTES command along with a FILE_SET_POSITION parameter value of FILE_IDENTIFIER_POSITION.

User Action: Omit the FILE_SEQUENCE_NUMBER parameter from the CHANGE_TAPE_LABEL_ATTRIBUTES command.

--WARNING AM 1477-- File {file} : Parameter FILE_IDENTIFIER is ignored for FILE_SET_POSITION = FILE_SEQUENCE_POSITION and REWRITE_LABELS = FALSE.

Condition Identifier: AME\$FILE_IDENTIFIER_IGNORED Code: AM 1477

Description: A FILE_IDENTIFIER parameter has been specified on the CHANGE_TAPE_LABEL_ATTRIBUTES command along with a REWRITE_LABELS parameter value of FALSE.

User Action: Omit the FILE_IDENTIFIER parameter from the CHANGE_TAPE_LABEL_ATTRIBUTES command or specify a REWRITE_LABELS parameter value of TRUE.

--WARNING AM 1478-- File {file} : Parameter GENERATION_NUMBER is ignored for FILE_SET_POSITION = FILE_SEQUENCE_POSITION and REWRITE_LABELS = FALSE.

Condition Identifier: AME\$GENERATION_NUMBER_IGNORED Code: AM 1478

Description: A GENERATION_NUMBER parameter has been specified on the CHANGE_TAPE_LABEL_ATTRIBUTES command along with a REWRITE_LABELS parameter value of FALSE.

User Action: Omit the GENERATION_NUMBER parameter from the CHANGE_TAPE_LABEL_ATTRIBUTES command or specify a REWRITE_LABELS parameter value of TRUE.

--ERROR AM 1479-- File {file} : Parameter FILE_IDENTIFIER must be specified for FILE_SET_POSITION = FILE_IDENTIFIER_POSITION.

Condition Identifier: AME\$FILE_IDENTIFIER_REQUIRED Code: AM 1479

Description: A FILE_IDENTIFIER parameter must be specified on the CHANGE_TAPE_LABEL_ATTRIBUTES command if a FILE_SET_POSITION value of FILE_IDENTIFIER_POSITION is specified.

User Action: Include a FILE_IDENTIFIER parameter on the CHANGE_TAPE_LABEL_ATTRIBUTES command.

--ERROR AM 1480-- File {file} : Parameter FILE_SEQUENCE_NUMBER must be specified for FILE_SET_POSITION = FILE_SEQUENCE_POSITION.

Condition Identifier: AME\$FILE_SEQ_NUMBER_REQUIRED Code: AM 1480

Description: A FILE_SEQUENCE_NUMBER parameter must be specified on the CHANGE_TAPE_LABEL_ATTRIBUTES command if a FILE_SET_POSITION value of FILE_SEQUENCE_POSITION is specified.

User Action: Include a FILE_SEQUENCE_NUMBER parameter on the CHANGE_TAPE_LABEL_ATTRIBUTES command.

--ERROR AM 1481-- File {file} : TAPE_FILE_SEQUENCE_NUMBER cannot be specified if TAPE_FILE_SET_POSITION.POSITION = FSC\$TAPE_FILE_IDENTIFIER_POSITION.

Condition Identifier: AME\$FILE_SEQ_NUMBER_ILLEGAL Code: AM 1481

Description: An fsc\$tape_file_sequence_number value is not allowed on an FSP\$OPEN_FILE tape file attachment if an fsc\$tape_file_set_position value of fsc\$tape_file_identifier_position is specified.

User Action: Omit the fsc\$tape_file_sequence_number entry from the tape file attachment.

--ERROR AM 1482-- File {file} : TAPE_FILE_IDENTIFIER, {text}, does not match TAPE_FILE_SET_POSITION.FILE_IDENTIFIER, {text}.

Condition Identifier: AME\$FILE_IDENTIFIER_MISMATCH Code: AM 1482

Description: The fse\$tape_file_identifier value specified on an FSP\$OPEN_FILE tape file attachment with a rewrite_labels value of FALSE differs from the file_identifier specified in the fse\$tape_file_set_position record.

User Action: Omit the fse\$tape_file_identifier entry from the tape file attachment.

--ERROR AM 1483-- File {file} : TAPE_GENERATION_NUMBER, {text}, does not match TAPE_FILE_SET_POSITION.GENERATION_NUMBER, {text}.

Condition Identifier: AME\$GENERATION_NUMBER_MISMATCH Code: AM 1483

Description: The fsc\$tape_generation_number value specified on an FSP\$OPEN_FILE tape file attachment with a rewrite_labels value of FALSE differs from the generation_number specified in the fsc\$tape_file_set_position record.

User Action: Omit the fsc\$tape_generation_number entry from the tape file attachment.

--ERROR AM 1484-- File {file} : TAPE_FILE_SEQUENCE_NUMBER, {text}, does not match TAPE_FILE_SET_POSITION.FILE_SEQUENCE_NUMBER, {text}.

Condition Identifier: AME\$FILE_SEQ_NUMBER_MISMATCH Code: AM 1484

Description: The fsc\$tape_sequence_number value specified on an FSP\$OPEN_FILE tape file attachment with a rewrite_labels value of FALSE differs from the file_sequence_number specified in the fse\$tape_file_set_position record.

User Action: Omit the fsc\$tape_file_sequence_number entry from the tape file attachment.

--ERROR AM 1485-- File {file} : {text} is not supported for ANSI labelled tape files.

Condition Identifier: AME\$UNSUPPORTED_TAPE_BT_RT Code: AM 1485

Description: The specified combination of block type and record type are not supported for ANSI labelled tape files.

User Action: Specify a different block type and record type combination.

--ERROR AM 1486-- File {file} : {text} is not implemented for ANSI labelled tape files.

Condition Identifier: AME\$UNIMPLEMENTED_TAPE_BT_RT Code: AM 1486

Description: The specified combination of block type and record type are not supported for ANSI labelled tape files.

User Action: Specify a different block type and record type combination.

--ERROR AM 1487-- File {file} : {text} is supported only for ANSI labelled tape files.

Condition Identifier: AME\$BT_RT_SUPP_ONLY_FOR_LABELED Code: AM 1487

Description: The block type/record type specified is only supported if LABEL_TYPE of the file is labelled. That is, the block type = US and record type = D (us/d, us/s) are only supported for labelled tape files.

User Action: Either change LABEL_TYPE of the file to labelled or specify a different block type and record type combination.

--ERROR AM 1503-- File {file} : Access to tape volume {text} not allowed.

Condition Identifier: AME\$ACCESSIBILITY_CONFLICT Code: AM 1503

Description: The file accessibility code specified for the ANSI file does not match the file accessibility code found in the VOL1 label on the tape volume.

User Action: Specify the correct file accessibility code for the tape volume.

--ERROR AM 1504-- File {file} : Specified ANSI file not found on the volume set.

Condition Identifier: AME\$FILE_NOT_IN_VOLUME_SET Code: AM 1504

Description: The specified ANSI file could not be found on the set of tape volumes that were provided.

User Action: Specify the correct identification for the desired ANSI file.

--ERROR AM 1505-- File {file} : Specified file set identifier of {text} conflicts with HDR1 file set identifier of {text}.

Condition Identifier: AME\$FILE_SET_ID_CONFLICT Code: AM 1505

Description: The specified file set identifier does not match the file set identifier in the HDR1 label of the ANSI file.

User Action: Specify the correct file set identifier for the ANSI file.

--ERROR AM 1507-- File {file} : Specified file sequence number is less than the file sequence number of the first ANSI file on the tape volume.

Condition Identifier: AME\$FSN_OUT_OF_SEQUENCE Code: AM 1507

Description: The file sequence number in the HDR1 label of the ANSI file at the current position on the tape is not one greater than the previous ANSI file on the tape. This can be caused by specifying the volumes in a multivolume set in the wrong order.

User Action: Specify the tape volumes in the correct order.

--ERROR AM 1508-- File {file} : Tape label {text} does not conform to ANSI standard format.

Condition Identifier: AME\$NON_ANSI_LABELS Code: AM 1508

Description: The content of the label block at the current position on the tape file does not conform to the format of an ANSI labelled tape file.

User Action: Labelled tapes containing non-ANSI standard labels can be accessed by an application with a label_type value of non_standard_labelled, a block_type of user_specified, and a record_type of undefined.

--ERROR AM 1509-- File {file} : File section number {text} accessed out of sequence.

Condition Identifier: AME\$SECTION_OUT_OF_SEQUENCE Code: AM 1509

Description: The file section number in the HDR1 label at the current position on the tape file is out of sequence. This can be caused by specifying the volumes in a multivolume set in the wrong order.

User Action: Specify the tape volumes in the correct order.

--ERROR AM 1510-- File {file} : {text} operation not allowed on an ANSI labelled tape file.

Condition Identifier: AME\$IMPROPER_LABELLED_TAPE_OP Code: AM 1510

Description: This operation is not allowed on an ANSI standard labelled tape file.

User Action: Modify the program to omit the operation if the label_type of the file is labelled.

--ERROR AM 1511-- File {file} : {text} with SKIP_UNIT of TAPEMARKS not allowed on an ANSI labelled tape file.

Condition Identifier: AME\$IMPROPER_LABELLED_SKIP_UNIT Code: AM 1511

Description: A skip unit of tapemarks is not allowed on an ANSI labelled tape file.

User Action: Modify the program to omit the AMP\$SKIP if the label_type of the file is labelled.

--ERROR AM 1512-- File {file} : Specified file sequence number {text} is greater than that of the last ANSI file on the volume set plus one. Use file_set_position of end_of_set to append a file to the volume set.

Condition Identifier: AME\$SPEC_FSN_OUT_OF_SEQ Code: AM 1512

Description: The specified file sequence number exceeds the file sequence number of the last ANSI file in the file set by more than one.

User Action: Change the file_sequence_number to be one greater than the last ANSI file on the file set or specify a file_set_position of end_of_set.

--ERROR AM 1513-- File {file} : SKIP backward encountered beginning of volume but NOT beginning of information.

Condition Identifier: AME\$SKIP_ENCOUNTERED_BOT Code: AM 1513

Description: An AMP\$SKIP on a multi-volume ANSI file with a skip_direction of backwards, encountered the beginning of the current volume.

User Action: Modify the program to use an AMP\$REWIND followed by an AMP\$SKIP with a skip_direction of forward in order to position the tape to the desired position.

--ERROR AM 1514-- File {file} : Improper open position for ANSI labelled tape file.

Condition Identifier: AME\$IMPROPER_OPEN_POSITION Code: AM 1514

Description: The specified open_position is not allowed on an ANSI labelled tape file.

User Action: Use an open_position of either \$BOI or \$EOI.

--ERROR AM 1515-- File {file} : {text} encountered end of set on an ANSI labelled tape file (more than 2 contiguous tapemarks).

Condition Identifier: AME\$3_CONTIGUOUS_TAPEMARKS Code: AM 1515

Description: Three contiguous tapemarks have been encountered on a labelled tape file.

User Action: Labelled tapes which contain embedded tapemarks can be accessed by an application with a label_type value of non_standard_labelled, a block_type of user_specified and a record_type of undefined.

--ERROR AM 1516-- File {file} : Non zero buffer offset length not implemented for ANSI labelled tape files.

Condition Identifier: AME\$UNIMPLEMENTED_BUFFER_OFFSET Code: AM 1516

Description: A nonzero buffer offset was encountered in the HDR2 label of an ANSI labelled tape file. NOS/VE does not support buffer offsets.

User Action: Labelled tapes written with nonzero buffer offsets can be accessed by an application with a label_type of non_standard_labelled, a block_type of user_specified, and a record_type of undefined.

--ERROR AM 1517-- File {file} : {text} operation on a blank labelled or non NOS/VE ANSI labelled tape file requires REWRITE_LABELS = TRUE.

Condition Identifier: AME\$REWRITE_LABELS_CONFLICT Code: AM 1517

Description: Writing on a blank labelled or non-NOS/VE labelled tape file is not allowed without specifying a rewrite_labels value of true.

User Action: Specify a REWRITE_LABELS value of TRUE.

--ERROR AM 1905-- File {file} : NIL STRUCTURE POINTER returned by {text}.

Condition Identifier: AME\$NIL_STRUCTURE_POINTER Code: AM 1905

Description: No data structure is defined for the FAP. Before fetching a pointer to the FAP data structure, you must define the structure with an AMP\$STORE_FAP_POINTER call.

User Action: Issue an AMP\$STORE_FAP_POINTER call when the file open call is passed to the FAP.

--CATASTROPHIC AP 0-- APL System ABORT due to internal error - {text}.

Condition Identifier: APE\$APL_SYSTEM_ABORT Code: AP 0

Description: If the APL system detects a problem that may result in the active workspace becoming unusable, it is reported as this error. These errors (except for DOUBLE INTERRUPT) are the result of problems in the APL system, operating system, or the computer equipment. Follow site-defined procedures for reporting these problems. If one of these errors occurs, the APL system attempts to continue in the hopes that the workspace is not damaged. If ten of these errors are detected in the same active workspace, APL will abort with an abnormal status. Therefore, do not replace an existing permanent file copy of the workspace with the active workspace. Instead, save the active workspace as a new permanent file. Then copy into a clearws the objects from the saved workspace. In most cases, any damaged objects will not be transferred. Any unrecoverable objects may then be restored from the original saved workspace.

If the error is due to DOUBLE INTERRUPT, this means that the APL system was interrupted a second time before the first interrupt could be processed. Some calculations may take a long time to do.

APL will process interrupts normally only after a primitive function finishes its execution. But if a second interrupt occurs, and over 100 milliseconds of CPU time has been used by the job (both task and monitor time) since the last unprocessed interrupt, APL terminates whatever is executing immediately. This may damage the active workspace. Further error reports with id=530000 may result due to this damage. (In a system which allows an address space of 2 billion bytes, it is possible for a primitive function to take a considerable amount of time to finish. Yet, most uses take very little time. There is a significant cost that would result from these functions detecting that an interrupt occurred while they were executing.)

When APL processes an interrupt, it generates an error. Your program may process this error using quad-trap and it may not respond to you due to a bug in your program or some other reason. In this event, you may: 1) wait in the hopes that your program will eventually respond; 2) terminate your job; or 3) enter enough interrupts to cause the APL system to abort.

User Action: Take appropriate corrective action.

--FATAL AP 1-- {text} is not an APL Workspace.

Condition Identifier: APE\$NOT_A_WORKSPACE Code: AP 1

Description: An attempt was made to load or copy from a file that is not recognizable as a workspace file. Workspace files have the following attributes associated with them.

```
FILE_ORGANIZATION = BYTE_ADDRESSABLE
FILE_STRUCTURE = DATA
FILE_CONTENTS = OBJECT
FILE_PROCESSOR = APL
```

A file without these attributes is not recognized as a workspace by the APL system.

Workspace files are highly structured. If a workspace file is changed outside of APL or a nonworkspace file is given the attributes of a workspace file, then an attempt to use such a file as a workspace will have undefined results.

User Action: Take appropriate corrective action.

--FATAL AP 2-- User Requested ABORT,{text}.

Condition Identifier: APE\$USER_REQUESTED_ABORT Code: AP 2

Description: The user aborted the task by executing the special function ABORT. The text is obtained from the argument to the ABORT function.

User Action: Take appropriate corrective action.

--ERROR AP 3-- Wrong type file ({text}) operation on {text}.

Condition Identifier: APE\$WRONG_TYPE_FILE Code: AP 3

Description: The user tried to do an operation on file local_file_name that is not allowed. For example, FREAD is not allowed on Coded files and CFREAD is not allowed on APL Structured files. operation is the operation attempted.

User Action: Take appropriate corrective action.

--ERROR AP 4-- File damage due to invalid {text} on {text}.

Condition Identifier: APE\$FILE_DAMAGE Code: AP 4

Description: Damage to an APL structured file was detected. reason may be one of the following

- **HEADER**
An invalid file header was detected. Much, if not all, information in the file is lost. Do not do any FWRITE, FRDEL, or FPACK operations to this file.
- **CHECKSUM DATA_TYPE RANK ELECNT**
A specific record was found to be unreadable because the record checksum did not match or the fields of the record were inconsistent.
- **RECORD_NUMBER**
The directory entry for this record does not correspond to the record. Many other records may be unreadable. Do not do any FWRITE, FRDEL, or FPACK operations on this file.

User Action: Take appropriate corrective action.

--ERROR AP 5-- File {text} exceeded on {text}.

Condition Identifier: APE\$FILE_ERROR Code: AP 5

Description: The APL structured file specified by local_file_name could not be enlarged enough to do the FWRITE or FPACK operation.

User Action: Take appropriate corrective action.

--ERROR AP 6-- {text} not allowed on {text}.

Condition Identifier: APE\$FILE_CHANGE_ERROR Code: AP 6

Description: Write access is required to do the operation on the file. The file should be reattached with ACCESS_MODE=ALL.

User Action: Take appropriate corrective action.

--ERROR AP 7-- Conversion error - {text}.

Condition Identifier: APE\$CONVERSION_ERROR Code: AP 7

Description: A problem was encountered when converting an APL2 workspace or structured file. Check the section on the CONVERT_APL2WS workspace.

User Action: Take appropriate corrective action.

--ERROR AP 8-- Workspace too big to save.

Condition Identifier: APE\$WORKSPACE_TOO_BIG Code: AP 8

Description: An attempt was made to save the active workspace as a permanent file and the new file would exceed 2,147,483,648 bytes. Active workspaces consist of several segments that must be combined into one segment to be saved as a file.

User Action: Take appropriate corrective action.

--FATAL AP 9-- Workspace is damaged - {text}.

Condition Identifier: APE\$WS_DAMAGED Code: AP 9

Description: The saved workspace file specified in the load or copy operation has an invalid checksum. Either the file has been damaged or the file is not a saved workspace even though it has the file attributes of a workspace.

User Action: Take appropriate corrective action.

--ERROR AP 10-- FUTIL error in {text}:{text}.

Condition Identifier: APE\$FUTIL_ERROR Code: AP 10

Description:

- rtn-name
The FUTIL routine in which the error was detected.
- err-msg
The error message.

User Action: Take appropriate corrective action.

--ERROR AP 11-- Error detected by Shared Variable Processor -{text}.

Condition Identifier: APE\$SHARED_VARIABLE_ERROR Code: AP 11

Description:

User Action: Take appropriate corrective action.

--ERROR AV 1-- An account name must be specified when a project name is specified.

Condition Identifier: AVE\$ACCOUNT_MUST_BE_SPECIFIED Code: AV 1

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 3-- {text} was called when the file was locked for read access.

Condition Identifier: AVE\$CONFLICTING_LOCK Code: AV 3

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5-- An internal error was detected in the index records for the template file.

Condition Identifier: AVE\$CORRUPTED_INDEX_RECORD Code: AV 5

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 7-- An internal error prevented {text} from retrieving the value of the variable {text} from the sequence {text}.

Condition Identifier: AVE\$CORRUPTED_SEQUENCE Code: AV 7

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 9-- Only a time may be specified.

Condition Identifier: AVE\$DATE_NOT_ALLOWED Code: AV 9

Description: A value was specified that included a date for a validation field which only allows times.

User Action: Specify a value that does not include a date.

--ERROR AV 11-- A date must be specified.

Condition Identifier: AVE\$DATE_REQUIRED Code: AV 11

Description: A value was specified that did not include a date for a validation field which requires a date.

User Action: Specify a value that includes a date.

--ERROR AV 13-- A date time range is not allowed.

Condition Identifier: AVE\$DATE_TIME_RANGE_NOT_ALLOWED Code: AV 13

Description: A range of date time values was specified for a validation field which does not allow a range.

User Action: Specify a single date time value.

--ERROR AV 15-- A date time range is required.

Condition Identifier: AVE\$DATE_TIME_RANGE_REQUIRED Code: AV 15

Description: A single date time value was specified for a validation field that requires a range.

User Action: Specify a range of date time values.

--ERROR AV 17-- A description record with the name {text} already exists.

Condition Identifier: AVE\$DESC_RECORD_ALREADY_EXISTS Code: AV 17

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 19-- The description record directory is full.

Condition Identifier: AVE\$DESC_RECORD_DIRECTORY_FULL Code: AV 19

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 21-- The end of the file was encountered.

Condition Identifier: AVE\$END_OF_TEMPLATE_FILE Code: AV 21

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 23-- A field with the name {text} already exists in description record {text}.

Condition Identifier: AVE\$FIELD_ALREADY_EXISTS Code: AV 23

Description: An attempt was made to create a validation field with a name that conflicts with an existing validation field.

User Action: Select a different name for the validation field being created.

--ERROR AV 25-- The field directory is full for description record {text}.

Condition Identifier: AVE\$FIELD_DIRECTORY_FULL Code: AV 25

Description: An attempt was made to create a validation field, but the maximum number of validation fields has already been created.

User Action: No further validation fields may be defined.

--ERROR AV 27-- A field with the name {text} was deleted at {text}.

Condition Identifier: AVE\$FIELD_WAS_DELETED Code: AV 27

Description: An attempt was made to reference a validation field that has been deleted.

User Action: Remove references to the validation field or restore the validation field.

--ERROR AV 29-- The value specified for the DEPTH parameter ({text}) is greater than the index depth of the file ({text}).

Condition Identifier: AVE\$INCORRECT_DEPTH Code: AV 29

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 31-- The password expiration interval must be less than or equal to the maximum expiration interval.

Condition Identifier: AVE\$INCORRECT_EXP_INTERVAL Code: AV 31

Description: The specified expiration interval is greater than the maximum interval allowed for this user.

User Action: Select an expiration interval that is less than or equal to the maximum allowed for this user, or change the maximum expiration interval for this user.

--ERROR AV 33-- The password expiration date is beyond the maximum expiration interval.

Condition Identifier: AVE\$INCORRECT_EXPIRATION_DATE Code: AV 33

Description: The specified expiration date is beyond than the maximum interval allowed for this user.

User Action: Select an expiration date within the maximum expiration interval allowed for this user, or change the maximum expiration interval for this user.

--ERROR AV 35-- "{text}" is not a correct file reference.

Condition Identifier: AVE\$INCORRECT_FILE_REFERENCE Code: AV 35

Description: The value specified for a file type validation field is not a valid file reference.

User Action: Specify a valid file reference, or a string that contains a valid file reference.

--ERROR AV 37-- The job warning limit must be less than or equal to the job maximum limit.

Condition Identifier: AVE\$INCORRECT_JOB_LIMITS Code: AV 37

Description: The value specified for the job warning limit is greater than the job maximum limit.

User Action: Specify a job warning limit value that is less than or equal to the job maximum limit, or change the job maximum limit.

--ERROR AV 39-- The minimum ring must be less than or equal to the nominal ring.

Condition Identifier: AVE\$INCORRECT_RING_PRIVILEGES Code: AV 39

Description: The value specified for the minimum ring is greater than the nominal ring for this user.

User Action: Specify a minimum ring which is less than or equal to the nominal ring, or change the nominal ring for this user.

--ERROR AV 41-- {text} fields have not been implemented yet.

Condition Identifier: AVE\$KIND_NOT_IMPLEMENTED Code: AV 41

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 43-- The {text} kind does not match the {text} kind.

Condition Identifier: AVE\$KINDS_DO_NOT_MATCH Code: AV 43

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 45-- The {text} parameter passed on the call to {text} is too small.

Condition Identifier: AVE\$NAME_LIST_TOO_SMALL Code: AV 45

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 47-- The WORK_AREA parameter passed on the call to {text} is NIL.

Condition Identifier: AVE\$NO_WORK_AREA Code: AV 47

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 49-- {text} cannot be restored because it is not a deleted field.

Condition Identifier: AVE\$NOT_A_DELETED_FIELD Code: AV 49

Description: An attempt was made to restore a validation field that has not been deleted.

User Action: Only previously deleted validation fields may be restored.

--ERROR AV 51-- {file} is not a template file.

Condition Identifier: AVE\$NOT_A_TEMPLATE_FILE Code: AV 51

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 53-- The default {text} job class must appear in the list of valid job classes.

Condition Identifier: AVE\$NOT_VALIDATED_FOR_DEFAULT Code: AV 53

Description: The value specified for the interactive or batch default job class is not in the list of valid job classes for this user.

User Action: Specify a job class which is in the list of valid job classes for this user, or add the job class to the list of valid job classes for this user.

--ERROR AV 55-- {text} is not in the range {text} to {text}.

Condition Identifier: AVE\$OUT_OF_RANGE Code: AV 55

Description: The value specified for a validation field is outside the range allowed for the field.

User Action: Specify a value that is inside the allowed range for the validation field.

--ERROR AV 57-- A record with the key {text} already exists.

Condition Identifier: AVE\$RECORD_ALREADY_EXISTS Code: AV 57

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 59-- An attempt was made to construct a record that is too large.

Condition Identifier: AVE\$RECORD_TOO_LARGE Code: AV 59

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 61-- The string length must be less than or equal to {text}.

Condition Identifier: AVE\$STRING_TOO_LONG Code: AV 61

Description: The value specified for a string type validation field is longer than allowed for the validation field.

User Action: Specify a value that is within the allowed length for the validation field.

--ERROR AV 63-- The string length must be greater than or equal to {text}.

Condition Identifier: AVE\$STRING_TOO_SHORT Code: AV 63

Description: The value specified for a string type validation field is shorter than allowed for the validation field.

User Action: Specify a value that is within the allowed length for the validation field.

--ERROR AV 65-- The template file is full.

Condition Identifier: AVE\$TEMPLATE_FILE_FULL Code: AV 65

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 67-- Only a date may be specified.

Condition Identifier: AVE\$TIME_NOT_ALLOWED Code: AV 67

Description: A value was specified that included a time for a validation field which only allows dates.

User Action: Specify a value that does not include a time.

--ERROR AV 69-- A time must be specified.

Condition Identifier: AVE\$TIME_REQUIRED Code: AV 69

Description: A value was specified that did not include a time for a validation field which requires a time.

User Action: Specify a value that includes a time.

--ERROR AV 71-- At least {text} names must be specified.

Condition Identifier: AVE\$TOO_FEW_NAMES Code: AV 71

Description: The value specified for a name type validation field contains fewer names than allowed for the validation field.

User Action: Specify a value that contains the allowed number of names for the validation field.

--ERROR AV 73-- Only {text} names may be specified.

Condition Identifier: AVE\$TOO_MANY_NAMES Code: AV 73

Description: The value specified for a name type validation field contains more names than allowed for the validation field.

User Action: Specify a value that contains the allowed number of names for the validation field.

--ERROR AV 75-- The {text} pointer in the value for field {text} is NIL.

Condition Identifier: AVE\$UNEXPECTED_NIL_POINTER Code: AV 75

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 77-- A description record with the name {text} could not be found.

Condition Identifier: AVE\$UNKNOWN_DESCRIPTION_RECORD Code: AV 77

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 79-- A field with the name {text} could not be found.

Condition Identifier: AVE\$UNKNOWN_FIELD Code: AV 79

Description: An attempt was made to reference a validation field that does not exist.

User Action: Verify the name of the field being referenced.

--ERROR AV 81-- An unknown field kind was detected by procedure {text}.

Condition Identifier: AVE\$UNKNOWN_FIELD_KIND Code: AV 81

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 83-- A record with the key {text} could not be found.

Condition Identifier: AVE\$UNKNOWN_RECORD Code: AV 83

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 85-- The file must be locked for update access before calling {text}.

Condition Identifier: AVE\$UPDATE_LOCK_REQUIRED Code: AV 85

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 87-- The utility information sequence passed on the call to {text} is too small.

Condition Identifier: AVE\$UTILITY_INFO_TOO_SMALL Code: AV 87

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 89-- {text} is not a name.

Condition Identifier: AVE\$VALUE_IS_NOT_A_NAME Code: AV 89

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 91-- The work area provided on the call to {text} is full.

Condition Identifier: AVE\$WORK_AREA_FULL Code: AV 91

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 101-- The account {text} already exists.

Condition Identifier: AVE\$ACCOUNT_ALREADY_EXISTS Code: AV 101

Description: An attempt was made to create an account validation record that already exists.

User Action: Verify that the correct account name was specified.

--ERROR AV 103-- The account {text} does not exist.

Condition Identifier: AVE\$ACCOUNT_DOES_NOT_EXIST Code: AV 103

Description: A reference was made to an account that does not exist.

User Action: Verify that the correct account name was specified.

--WARNING AV 104-- The account {text} does not exist.

Condition Identifier: AVE\$ACCOUNT_DOES_NOT_EXIST_WARN Code: AV 104

Description: The account specified for the validation field does not exist, however the field has been set to the specified value.

User Action: Verify that the correct account name was specified. If the account name is correct, create the validations for the specified account.

--ERROR AV 106-- The login validation information for the current executing account is unavailable.

Condition Identifier: AVE\$ACCOUNT_INFO_NOT_FOUND Code: AV 106

Description: An attempt was made to retrieve an account validation field for the executing job, but the system's validation level is user.

User Action: The validation level must be set to account or project in order to be able to access account validation information for a job.

--ERROR AV 109-- The user {text} is already a member of account {text}.

Condition Identifier: AVE\$ACCT_MEMBER_ALREADY_EXISTS Code: AV 109

Description: An attempt was made to create an account member validation record that already exists.

User Action: Verify that the correct account name and user name were specified.

--ERROR AV 112-- The user {text} is not a member of account {text}.

Condition Identifier: AVE\$ACCT_MEMBER_DOES_NOT_EXIST Code: AV 112

Description: A reference was made to an account member that does not exist.

User Action: Verify that the correct account name and user name were specified.

--WARNING AV 113-- The user {text} is not a member of account {text}.

Condition Identifier: AVE\$ACC_MEM_DOES_NOT_EXIST_WARN Code: AV 113

Description: The user is not a member of the specified account, however the field has been set to the specified value.

User Action: Verify that the correct account name was specified. If the account name is correct, create account member validations for the user.

--ERROR AV 115-- The login validation information for the current executing account member is unavailable.

Condition Identifier: AVE\$ACCT_MEMBER_INFO_NOT_FOUND Code: AV 115

Description: An attempt was made to retrieve an account member validation field for the executing job, but the system's validation level is user.

User Action: The validation level must be set to account or project in order to be able to access account member validation information for a job.

--ERROR AV 118-- User not valid for job class {text}.

Condition Identifier: AVE\$BAD_JOB_CLASS Code: AV 118

Description: An attempt was made to access a job class that does not exist or for which the user is not validated.

User Action: Verify that the specified job class is correct.

--ERROR AV 121-- User not valid for access to ring {text}.

Condition Identifier: AVE\$BAD_RING Code: AV 121

Description: An attempt was made to run a job at a ring that is below the user's minimum ring.

User Action: Verify that the specified ring is correct.

--ERROR AV 124-- User specified a {text} job maximum limit that is greater than the user's validation.

Condition Identifier: AVE\$BAD_USER_SPECIFIED_JOB_MAX Code: AV 124

Description: On the LOGIN command, the user specified a job maximum value that is greater than the user's validation.

User Action: Specify a job maximum limit that is within the user's allowed validations.

--ERROR AV 127-- Incorrect user validation information.

Condition Identifier: AVE\$BAD_USER_VALIDATION_INFO Code: AV 127

Description: An incorrect user name or password has been specified on the LOGIN command.

User Action: Correct the job's LOGIN command.

--ERROR AV 128-- Activate or Deactivate administrator can not be called when more than 1 task is active for the job.

Condition Identifier: AVE\$CALL_INVALID_TOO_MANY_TASKS Code: AV 128

Description: An attempt was made to change the system or family administration status when more than one task is active for the job.

User Action: This may be done only when no other tasks or utilities are active for a job.

--ERROR AV 130-- Deletion of the currently executing {text} is not allowed.

Condition Identifier: AVE\$CAN_NOT_DELETE_CURRENT Code: AV 130

Description: An attempt was made to delete the executing validation record.

User Action: Users may not delete their own validations.

--ERROR AV 133-- A specific entry cannot be deleted from a list which currently contains the value ALL.

Condition Identifier: AVE\$CANNOT_DELETE_NAME_FROM_ALL Code: AV 133

Description: An attempt was made to delete a specific name from a name list that contains the value ALL.

User Action: The list of desired names must be specified individually.

--ERROR AV 134-- {text} is not a capability or is not supported for this command at this time.

Condition Identifier: AVE\$CAPABILITY_NOT_SUPPORTED Code: AV 134

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 136-- Only a system administrator may set a new password without supplying the current password.

Condition Identifier: AVE\$CAN_NOT_SET_NEW_WITHOUT_OLD Code: AV 136

Description: A family administrator attempted to specify a new validation file password without supplying the current password for the validation file.

User Action: Supply the current validation file password along with the new validation file password.

--ERROR AV 137-- Activate or Deactivate administrator not be executed from within the system job.

Condition Identifier: AVE\$CALL_NOT_VALID_FROM_SYS_JOB Code: AV 137

Description: An attempt was made to change the administration status of the system job.

User Action: Changing the administration status of the system job is not allowed.

--ERROR AV 138-- The command {text} already exists for field {text}.

Condition Identifier: AVE\$CMD_ALREADY_IN_CMD_TABLE Code: AV 138

Description: The value specified (or defaulted) for the CHANGE_COMMAND_NAMES or DISPLAY_COMMAND_NAMES parameter conflicts with a command (or abbreviation) being used for another field.

User Action: Specify a value that does not conflict with the commands being used for other fields.

--ERROR AV 139-- An encrypted password value may only be specified on a source run.

Condition Identifier: AVE\$ENCRYPTED_PW_NOT_ALLOWED Code: AV 139

Description: A value was specified for the ENCRYPTED_PASSWORD parameter.

User Action: Only unencrypted passwords are allowed at this time. Source runs are not implemented.

--ERROR AV 140-- Deletion of files is not allowed when deleting a user from a local validation file.

Condition Identifier: AVE\$DELETE_FILES_NOT_ALLOWED Code: AV 140

Description: A value of TRUE was specified for the DELETE_FILES parameter when deleting a user from a local validation file.

User Action: Specify a value of FALSE for the DELETE_FILES parameter.

--ERROR AV 142-- The command table array supplied ({text} entries) is not big enough to receive the command table ({text} entries).

Condition Identifier: AVE\$INCORRECT_CMD_TABLE_SIZE Code: AV 142

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 145-- Field {text} is not a {text} type field.

Condition Identifier: AVE\$INCORRECT_KIND Code: AV 145

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 148-- User does not have the required authority to perform this request. {text}

Condition Identifier: AVE\$INSUFFICIENT_AUTHORITY Code: AV 148

Description: An attempt was made to perform a validation function for which the user is not validated.

User Action: A user may only perform validation functions for which they are validated.

--ERROR AV 151-- The password returned by AVP\$PROCESS_PASSWORD_ATTRIBUTES can not be a null name.

Condition Identifier: AVE\$INVALID_PASSWORD_FROM_HOOK Code: AV 151

Description: An internal error has occurred in site modified code.

User Action: Provide a site analyst with appropriate information.

--WARNING AV 153-- A master catalog for {text} already exists.

Condition Identifier: AVE\$MASTER_CATALOG_EXISTS Code: AV 153

Description: A master catalog with the same name as the user just created already exists.

User Action: Determine that the correct user name was used when creating the user. If the correct user name was used, determine if the user just created should have access to the files in this master catalog and take appropriate action. If an incorrect user name was used, delete the user (using the DELETE_USER command with a value of FALSE for the DELETE_FILES parameter) to undo the action of the CREATE_USER command.

--ERROR AV 154-- The user {text} is neither a member of account {text} nor a member of project {text} within account {text}.

Condition Identifier: AVE\$MEMBER_DOES_NOT_EXIST Code: AV 154

Description: An attempt was made to access a project for which the user is not validated for access.

User Action: Verify that the correct account name, project name, and user name were specified.

--WARNING AV 155-- The user {text} is neither a member of account {text} nor a member of project {text} within account {text}.

Condition Identifier: AVE\$MEMBER_DOES_NOT_EXIST_WARN Code: AV 155

Description: The user is not a member of the specified project; however, the field has been set to the specified value.

User Action: Verify that the account and project are correct. If they are correct, create project or account member validations for the user.

--ERROR AV 157-- User does not have required capability {text}.

Condition Identifier: AVE\$MISSING_REQUIRED_CAPABILITY Code: AV 157

Description: The user must have the specified capability to access NOS/VE as requested.

User Action: Only users who are validated for the specified capability may access NOS/VE as requested.

--ERROR AV 160-- Unable to find validation record information for {text}.

Condition Identifier: AVE\$MISSING_VAL_RECORD_INFO Code: AV 160

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 163-- System and/or family administrators must specify a password to access this validation file.

Condition Identifier: AVE\$MUST_SPECIFY_PASSWORD Code: AV 163

Description: A system or family administrator must specify a password to access any validation file that has a security password on it.

User Action: Specify the security password via the USE_VALIDATION_FILE command.

--ERROR AV 166-- No password exists.

Condition Identifier: AVE\$NO_PASSWORD_ON_FILE Code: AV 166

Description: A system or family administrator specified a validation file security password for a validation file that has no security password.

User Action: Execute the USE_VALIDATION_FILE command without specifying a security password.

--ERROR AV 167-- {text} is not allowed on the client mainframe.

Condition Identifier: AVE\$NOT_ALLOWED_ON_CLIENT Code: AV 167

Description: The requested action is currently not supported on a client mainframe.

User Action: Perform the action on the server mainframe.

--ERROR AV 169-- The password specified does not match the current password for this {text}.

Condition Identifier: AVE\$OLD_PASSWORD_NOT_VALID Code: AV 169

Description: An incorrect old password value was specified.

User Action: Verify that the old password value specified is correct.

--ERROR AV 172-- The user {text} is already a member of project {text} within account {text}.

Condition Identifier: AVE\$PROJ_MEMBER_ALREADY_EXISTS Code: AV 172

Description: An attempt was made to create a project member validation record that already exists.

User Action: Verify that the correct account name, project name, and user name were specified.

--ERROR AV 175-- The user {text} is not a member of project {text} within account {text}.

Condition Identifier: AVE\$PROJ_MEMBER_DOES_NOT_EXIST Code: AV 175

Description: A reference was made to a project member that does not exist.

User Action: Verify that the correct account name, project name, and user name were specified.

--ERROR AV 178-- The login validation information for the current executing project member is unavailable.

Condition Identifier: AVE\$PROJ_MEMBER_INFO_NOT_FOUND Code: AV 178

Description: An attempt was made to retrieve a project member validation field for the executing job, but the system's validation level is not project.

User Action: The validation level must be set to project to access project member validation information for a job.

--ERROR AV 180-- The project {text} within account {text} already exists.

Condition Identifier: AVE\$PROJECT_ALREADY_EXISTS Code: AV 180

Description: An attempt was made to create a project validation record that already exists.

User Action: Verify that the correct account name and project name were specified.

--ERROR AV 182-- The project {text} within account {text} does not exist.

Condition Identifier: AVE\$PROJECT_DOES_NOT_EXIST Code: AV 182

Description: A reference was made to a project that does not exist.

User Action: Verify that the correct account name and project name were specified.

--WARNING AV 183-- The project {text} within account {text} does not exist.

Condition Identifier: AVE\$PROJECT_DOES_NOT_EXIST_WARN Code: AV 183

Description: The project specified for the validation field does not exist, however the field has been set to the specified value.

User Action: Verify that the correct account name and project name were specified. If the project name is correct, create the validations for the specified project.

--ERROR AV 184-- The login validation information for the current executing project is unavailable.

Condition Identifier: AVE\$PROJECT_INFO_NOT_FOUND Code: AV 184

Description: An attempt was made to retrieve a project validation field for the executing job, but the system's validation level is not project.

User Action: The validation level must be set to project to access project validation information for a job.

--ERROR AV 187-- The reserved name {text} is not a valid name for creation of a {text}.

Condition Identifier: AVE\$RESERVED_NAME_NOT_VALID Code: AV 187

Description: An attempt was made to create a validation record of the type specified with a special name that is not allowed.

User Action: Select a different name.

--ERROR AV 190-- The {text} {text} total limit has been reached or exceeded.

Condition Identifier: AVE\$TOTAL_LIMIT_REACHED Code: AV 190

Description: The user has reached or exceeded the specified total limit.

User Action: Contact the family administrator to have the total limit raised or the accumulator reset.

--ERROR AV 193-- Unable to find field name for command {text}.

Condition Identifier: AVE\$UNABLE_TO_FIND_FIELD_NAME Code: AV 193

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 194-- An unknown validation record was specified on a request for validation information.

Condition Identifier: AVE\$UNKNOWN_VALIDATION_RECORD Code: AV 194

Description: An incorrect value was specified for the RECORD_LEVEL parameter when attempting to retrieve validation information about the currently executing job.

User Action: Specify a correct value for the RECORD_LEVEL parameter.

--ERROR AV 195-- The user {text} already exists.

Condition Identifier: AVE\$USER_ALREADY_EXISTS Code: AV 195

Description: The specified user has already been created.

User Action: Choose a different user name.

--ERROR AV 197-- The user {text} does not exist.

Condition Identifier: AVE\$USER_DOES_NOT_EXIST Code: AV 197

Description: The specified user does not exist.

User Action: Verify that the correct user name was specified.

--ERROR AV 199-- The login validation information for the current executing user is unavailable.

Condition Identifier: AVE\$USER_INFO_NOT_FOUND Code: AV 199

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--WARNING AV 311-- The master catalog for user {text} in family {text} already exists.

Condition Identifier: AVE\$CATALOG_ALREADY_EXISTS Code: AV 311

Description: The CREATE_FAMILY command attempted to create a master catalog for the specified user, but the master catalog already exists.

User Action: Verify that the correct values were specified on the CREATE_FAMILY command. If they were correct, no action is required.

--WARNING AV 312-- Validation information for user {text} in family {text} already exists.

Condition Identifier: AVE\$USER_VALIDATION_INFO_EXISTS Code: AV 312

Description: The CREATE_FAMILY command attempted to create the validations for the specified user, but a validation record for this user already exists. Note: The validation record is NOT updated to reflect the values specified on the CREATE_FAMILY command.

User Action: Verify that the correct values were specified on the CREATE_FAMILY command. If they were, no action is required.

--INFORMATIVE AV 502-- User requested logout during account/project entry.

Condition Identifier: AVE\$USER_REQUESTS_LOGOUT Code: AV 502

Description: User has requested logout in response to the prompt Enter Account name and Project name.

User Action: None. User has been logged out.

--ERROR AV 701-- Segment pointer specified is NIL. ({text} request)

Condition Identifier: AVE\$NIL_SEGMENT Code: AV 701

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 702-- Specified segment too small for Data Base Manager. ({text} request)

Condition Identifier: AVE\$SEGMENT_TOO_SMALL Code: AV 702

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 703-- Version {text} of the \$FAMILY_USERS file is unknown.

Condition Identifier: AVE\$UNKNOWN_VAL_FILE_VERSION Code: AV 703

Description: The version identifier on the validation file has been lost or identifies a version that is no longer supported. This can occur if an attempt is made to upgrade to NOS/VE 1.3.1 from a pre NOS/VE 1.2.1 system level.

User Action: Upgrade to a NOS/VE 1.2.1 or later system before attempting to upgrade to NOS/VE 1.3.1.

--WARNING AV 704-- An error occurred while opening the validation file for family {text}.

Condition Identifier: AVE\$OPEN_VAL_FILE_ERROR Code: AV 704

Description: NOS/VE was unable to open the validation file for the specified family. The validation file has been damaged and may need to be reloaded.

User Action: Not applicable.

--WARNING AV 705-- An error occurred while reading a user record for family {text}.

Condition Identifier: AVE\$ACCESS_USER_REC_ERROR Code: AV 705

Description: NOS/VE is unable to read user records from the validation file for the specified family during deadstart. The validation file has been damaged and may need to be reloaded.

User Action: Not applicable.

--WARNING AV 706-- An error occurred while creating user \$SYSTEM for family {text}.

Condition Identifier: AVE\$CANNOT_CREATE_SYSTEM_USER Code: AV 706

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--WARNING AV 707-- An error occurred while converting the validation file for family {text}.

Condition Identifier: AVE\$CANNOT_CONVERT_VAL_FILE Code: AV 707

Description: NOS/VE is unable to convert the validation file for the specified family during deadstart. The validation file has been damaged and may need to be reloaded.

User Action: Not applicable.

--WARNING AV 708-- An error occurred while updating the validation file version for family {text}.

Condition Identifier: AVE\$CANNOT_UPDATE_VAL_VERSION Code: AV 708

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--WARNING AV 709-- The validation file for family {text} is being recreated.

Condition Identifier: AVE\$RECREATING_SYSTEM_FAMILY Code: AV 709

Description: The validation file for the specified family is being recreated in order to allow deadstart to complete.

User Action: Not applicable.

--INFORMATIVE AV 1000-- Changing account {text}.

Condition Identifier: AVE\$CHANGING_ACCOUNT Code: AV 1000

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1001-- Changing account member {text} in account {text}.

Condition Identifier: AVE\$CHANGING_ACCOUNT_MEMBER Code: AV 1001

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1002-- Changing project {text} in account {text}

Condition Identifier: AVE\$CHANGING_PROJECT Code: AV 1002

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1003-- Changing project member {text} in project {text} of account {text}.

Condition Identifier: AVE\$CHANGING_PROJECT_MEMBER Code: AV 1003

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1004-- Changing user {text}.

Condition Identifier: AVE\$CHANGING_USER Code: AV 1004

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1005-- Creating account {text}.

Condition Identifier: AVE\$CREATING_ACCOUNT Code: AV 1005

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1006-- Creating account member {text} in account {text}.

Condition Identifier: AVE\$CREATING_ACCOUNT_MEMBER Code: AV 1006

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1007-- Creating project {text} in account {text}

Condition Identifier: AVE\$CREATING_PROJECT Code: AV 1007

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1008-- Creating project member {text} in project {text} of account {text}.

Condition Identifier: AVE\$CREATING_PROJECT_MEMBER Code: AV 1008

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--INFORMATIVE AV 1009-- Creating user {text}.

Condition Identifier: AVE\$CREATING_USER Code: AV 1009

Description: This message is used by ADMINISTER_VALIDATIONS to inform the user what action is being performed.

User Action: None.

--WARNING AV 1010-- The user does not have enough authority to add or delete the following capabilities: {text}.

Condition Identifier: AVE\$CANNOT_CHANGE_CAPABILITY Code: AV 1010

Description: An attempt was made to add or delete a capability that the user is not authorized to add or delete.

User Action: Only users with the appropriate authority may add or delete the specified capability.

--ERROR AV 1015-- Deletion of all {text} is not allowed.

Condition Identifier: AVE\$CANNOT_DELETE_ALL Code: AV 1015

Description: An attempt was made to delete all user validation records.

User Action: User validation records must be deleted by explicitly specifying the user names.

--ERROR AV 1020-- The OLD_PASSWORD and NEW_PASSWORD parameters cannot be used if a value is specified for the ENCRYPTED_PASSWORD parameter.

Condition Identifier: AVE\$CANNOT_SUPPLY_OLD_OR_NEW_PW Code: AV 1020

Description: The ENCRYPTED_PASSWORD parameter was specified along with the OLD_PASSWORD and/or NEW_PASSWORD parameters.

User Action: Do not specify the ENCRYPTED_PASSWORD parameter because it is not supported at this time.

--ERROR AV 1030-- The requested operation conflicts with an active {text} subutility within this ADMV session.

Condition Identifier: AVE\$CONFLICTING_OPERATION Code: AV 1030

Description: An attempt was made to operate on a validation record that is in the process of being created or changed by an ADMINISTER_VALIDATIONS subutility, currently active in this ADMINISTER_VALIDATIONS session. For example, deleting a validation record while a create or change subutility is operating on the record is not allowed.

User Action: Exit the ADMINISTER_VALIDATIONS subutility that is creating or changing the record before attempting the operation.

--ERROR AV 1055-- {text} cannot be used when an ADMINISTER_VALIDATIONS subutility is active.

Condition Identifier: AVE\$NOT_ALLOWED_FROM_SUBUTILITY Code: AV 1055

Description: An attempt was made to enter the specified subutility when another ADMINISTER_VALIDATIONS subutility is already active.

User Action: Exit the active subutility before entering the specified subutility.

--ERROR AV 1060-- The keyword value {text} must be used alone for parameter {text}.

Condition Identifier: AVE\$MUST_BE_USED_ALONE Code: AV 1060

Description: The user specified a keyword value along with other values for the specified parameter.

User Action: Specify a keyword value or a list of values that does not contain a keyword.

--WARNING AV 1065-- System and family administrators must use the USE_VALIDATION_FILE command, specifying the correct password, in order to access the validation file for this family.

Condition Identifier: AVE\$MUST_EXECUTE_USEVF_CMD Code: AV 1065

Description: The system or family administrator has entered the ADMINISTER_VALIDATIONS utility on a family where the validation file is protected by a security password.

User Action: Execute the USE_VALIDATION_FILE command specifying the security password for the validation file.

--ERROR AV 1070-- A value must be specified for the NEW_PASSWORD parameter if a value is specified for the OLD_PASSWORD parameter.

Condition Identifier: AVE\$MUST_SPECIFY_NEW_PASSWORD Code: AV 1070

Description: An attempt was made to specify an old password without specifying a new password.

User Action: Do not specify the OLD_PASSWORD parameter without specifying the NEW_PASSWORD parameter.

--INFORMATIVE AV 1071-- There are no account validations.

Condition Identifier: AVE\$NO_ACCOUNTS Code: AV 1071

Description: An attempt was made to display account validations when no account validations exist in the validation file.

User Action: None.

--INFORMATIVE AV 1072-- There are no account members for account {text}.

Condition Identifier: AVE\$NO_ACCOUNT_MEMBERS Code: AV 1072

Description: An attempt was made to display account member validations when no account member validations exist in the validation file.

User Action: None.

--INFORMATIVE AV 1073-- There are no projects for account {text}.

Condition Identifier: AVE\$NO_PROJECTS Code: AV 1073

Description: An attempt was made to display project validations when no project validations exist in the validation file.

User Action: None.

--INFORMATIVE AV 1074-- There are no project members for project {text} in account {text}.

Condition Identifier: AVE\$NO_PROJECT_MEMBERS Code: AV 1074

Description: An attempt was made to display project member validations when no project member validations exist in the validation file.

User Action: None.

--ERROR AV 1075-- A validation file must be opened by executing the USE_VALIDATION_FILE command.

Condition Identifier: AVE\$NO_VALIDATION_FILE_OPEN Code: AV 1075

Description: A validation file has not been selected for use.

User Action: Execute the USE_VALIDATION_FILE command specifying the appropriate parameters.

--ERROR AV 1085-- {text} is not a defined capability.

Condition Identifier: AVE\$UNKNOWN_CAPABILITY Code: AV 1085

Description: An attempt was made to reference a capability that does not exist.

User Action: Verify that the capability name is correct.

--ERROR AV 1090-- {text} is not a recognized display option.

Condition Identifier: AVE\$UNKNOWN_DISPLAY_OPTION Code: AV 1090

Description: An unrecognizable display option was specified.

User Action: Verify that the display option is correct.

--WARNING AV 5121-- DELETED username {text}.

Condition Identifier: AVE\$DELETED_USERNAME Code: AV 5121

Description: The listed username was deleted from the NOS/VE \$validations file.

User Action: None.

--WARNING AV 5124-- The limit specified for installation accumulator {text} has been exceeded.

Condition Identifier: AVE\$EXCEEDED_LR Code: AV 5124

Description: The listed project is over its NOS limit for the listed accumulator.

User Action: Reset the NOS limit for the listed accumulator (only if the project number is desired in the NOS/VE validation file.)

--WARNING AV 5127-- The installation specified SRU limit has been exceeded.

Condition Identifier: AVE\$EXCEEDED_SIL Code: AV 5127

Description: The listed project is over its NOS installation SRU limit.

User Action: Reset the NOS installation SRU limit for the listed project (only if the project number is desired in the NOS/VE validation file.)

--WARNING AV 5130-- The master user specified SRU limit has been exceeded.

Condition Identifier: AVE\$EXCEEDED_SML Code: AV 5130

Description: The listed project is over its NOS master user SRU limit.

User Action: Contact the master user for the specified account, who may alter the limit.

--WARNING AV 5142-- IGNORED username {text} (see reason(s) listed above).

Condition Identifier: AVE\$IGNORED_USERNAME Code: AV 5142

Description: The listed NOS username was not moved from the NOS validation files to the NOS/VE validation file.

User Action: Follow the action shown for the previous message you received.

--WARNING AV 5148-- The password cannot be used on NOS/VE.

Condition Identifier: AVE\$INVALID_PASSWORD Code: AV 5148

Description: The format of the NOS password for the listed username is invalid for a NOS/VE password.

User Action: Use a NOS password that is of a valid format for a NOS/VE password (only if the username is desired in the NOS/VE validation file.)

--WARNING AV 5154-- Username {text} cannot be used on NOS/VE.

Condition Identifier: AVE\$INVALID_USERNAME Code: AV 5154

Description: The format of the NOS username listed is invalid for a NOS/VE username.

User Action: Use a NOS username that is of a valid format for a NOS/VE username (only if the username is desired in the NOS/VE validation file.)

--WARNING AV 5157-- Username does not have AW={text} set in its NOS validation.

Condition Identifier: AVE\$MISSING_REQUIRED_AW_BIT Code: AV 5157

Description: The NOS access word must have AW=CUCP and AW=CNVE set if the listed user name is to be moved to the NOS/VE validation file.

User Action: Set AW=CUCP and AW=CNVE if you want the user name moved from the NOS validation files to the NOS/VE validation file.

--WARNING AV 5166-- REJECTED updates to username {text} (see reason(s) listed above).

Condition Identifier: AVE\$REJECTED_UPDATES Code: AV 5166

Description: Updates to the NOS/VE validation file for the listed username did not occur.

User Action: Follow the action shown for the previous message you received.

--ERROR AV 5205-- The AVC\$BEGIN_ACCOUNT statistic must be the first statistic processed.

Condition Identifier: AVE\$BEGIN_ACCOUNT_NOT_FIRST Code: AV 5205

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5210-- The desired message is not in the message list.

Condition Identifier: AVE\$CANNOT_FIND_MESSAGE Code: AV 5210

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5215-- One or more statistics that are required for billing were not found in the accounting log.

Condition Identifier: AVE\$MISSING_REQUIRED_STATISTIC Code: AV 5215

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5225-- No space is available in the heap for more messages.

Condition Identifier: AVE\$NO_SPACE_FOR_MESSAGE Code: AV 5225

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5230-- An unknown NOS account file message was found in the message list.

Condition Identifier: AVE\$UNKNOWN_NOS_MESSAGE Code: AV 5230

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5301-- ALL must be used alone for parameter {text}.

Condition Identifier: AVE\$ALL_MUST_BE_ALONE_ANAJAL Code: AV 5301

Description: The keyword ALL was specified along with other values for the specified parameter.

User Action: Specify ALL or the set of desired values.

--ERROR AV 5302-- NONE must be used alone for parameter {text}.

Condition Identifier: AVE\$NONE_MUST_BE_ALONE_ANAJAL Code: AV 5302

Description: The keyword NONE was specified along with other values for the specified parameter.

User Action: Specify NONE or the set of desired values.

--ERROR AV 5303-- Application end statistic found for application {text}, without an application begin statistic.

Condition Identifier: AVE\$APPLICATION_NOT_BEGUN Code: AV 5303

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5305-- No space is available in the heap for more resource info records.

Condition Identifier: AVE\$NO_SPACE_FOR_RESOURCE_INFO Code: AV 5305

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5310-- No space is available in the heap for more application records.

Condition Identifier: AVE\$NO_SPACE_FOR_APP_INFO Code: AV 5310

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5315-- No space is available in the heap for more job info records.

Condition Identifier: AVE\$NO_SPACE_FOR_JOB_INFO Code: AV 5315

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5330-- Wrong token type found on {text}. Expected {text} found {text}.

Condition Identifier: AVE\$WRONG_TOKEN_TYPE Code: AV 5330

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5335-- The following required statistics were not found: {text}.

Condition Identifier: AVE\$REQUIRED_STATISTIC_MISSING Code: AV 5335

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5405-- The start of this account log is not within {text} hour(s) after the last account log processed.

Condition Identifier: AVE\$ALLOWABLE_INTERVAL_EXCEEDED Code: AV 5405

Description: Account logs must be analyzed in chronological order. When analyzing an account log, the interval from the end of the previously analyzed account log to the start of the current account log must be within the user defined allowable interval. The allowable interval time is set by the ALLOWABLE_INTERVAL parameter on the ANALYZE_ACCOUNT_LOG command call.

User Action: 1) Determine that the correct account log file is being used and, if not, use the correct account log. 2) If this is the correct account log, rerun ANALYZE_ACCOUNT_LOG with the appropriate ALLOWABLE_INTERVAL parameter value.

--ERROR AV 5410-- The number of storage hours is not known. The storage hours parameter must be used.

Condition Identifier: AVE\$UNKNOWN_STORAGE_HOURS Code: AV 5410

Description: The number of storage hours for permanent file billing is automatically calculated by ANALYZE_ACCOUNT_LOG (ANAAL) as the time (to the nearest hour) between each execution of EMIT_PERMANENT_FILE_STATISTICS (EMIPFS). This error occurs only if this is the first execution of EMIPFS or the resource usage file has been damaged in some way.

User Action: 1) If this is the first execution of EMIPFS, rerun ANAAL with the appropriate STORAGE_HOURS parameter. 2) If this is not the first execution of EMIPFS, provide your site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5415-- The active system {text} file can not be used as input to ANAAL. A terminated log must be used.

Condition Identifier: AVE\$MUST_USE_TERMINATED_LOG Code: AV 5415

Description: The input file entered for the ACCOUNT_LOG parameter on the ANALYZE_ACCOUNT_LOG command referenced the active system account log.

User Action: Use TERMINATE_LOG to terminate the account log to a separate file, then rerun ANALYZE_ACCOUNT_LOG using the terminated log as input.

--ERROR AV 5417-- The account log file: {text} is empty.

Condition Identifier: AVE\$EMPTY_ACCOUNT_LOG Code: AV 5417

Description: ANALYZE_ACCOUNT_LOG was unable to process the specified account log because it is empty.

User Action: Specify an account log that is not empty.

--INFORMATIVE AV 5420-- The last account log processed ended at: {text} {text}

Condition Identifier: AVE\$LAST_LOG_END Code: AV 5420

Description: This message records the date and time of the last record on the account log that was last processed by ANALYZE_ACCOUNT_LOG.

User Action: None.

--INFORMATIVE AV 5425-- The ALLOWABLE_INTERVAL value is {text} hour(s).

Condition Identifier: AVE\$ALLOWABLE_INTERVAL_VALUE Code: AV 5425

Description: This message records the allowable interval value specified on the call to ANALYZE_ACCOUNT_LOG.

User Action: None.

--INFORMATIVE AV 5430-- The account log currently being processed starts at: {text} {text}

Condition Identifier: AVE\$CURRENT_LOG_START Code: AV 5430

Description: This message records the date and time of the first record on the account log currently being processed.

User Action: None.

--INFORMATIVE AV 5435-- The account log currently being processed ends at: {text} {text}

Condition Identifier: AVE\$CURRENT_LOG_END Code: AV 5435

Description: This message records the date and time of the last record on the account log currently being processed.

User Action: None.

--WARNING AV 5440-- The last time that ANALYZE_ACCOUNT_LOG was run is not known. Processing order will not be checked.

Condition Identifier: AVE\$UNKNOWN_LAST_LOG_END Code: AV 5440

Description: This warning should only appear the first time that ANALYZE_ACCOUNT_LOG is executed.

User Action: If this is the first time that ANALYZE_ACCOUNT_LOG was executed, no action is required. If this is not the first time that ANALYZE_ACCOUNT_LOG was executed, provide your site analyst with appropriate information or submit a Programming System Report (PSR).

--WARNING AV 5445-- ANAJAL was unable to process job {text} for the following reason:

Condition Identifier: AVE\$ANAJAL_ABORTED_WARNING Code: AV 5445

Description: An error occurred while the system epilog was executing ANALYZE_JOB_ACCOUNT_LOG for the listed job.

User Action: Determine what caused the error to occur. 1) If the error is correctable: Correct the situation that caused the error. Use EXTRACT_JOB to get a copy of the binary log messages for the listed job. Rerun ANALYZE_JOB_ACCOUNT_LOG with the extracted job log as input. 2) If the error is not correctable: Provide the site analyst with appropriate information or submit a Programming System Report (PSR).

--INFORMATIVE AV 5450-- Total statistics read = {text}

Condition Identifier: AVE\$STATISTICS_READ Code: AV 5450

Description: This message records the total number of statistics read from the account log currently being processed.

User Action: None.

--WARNING AV 5455-- A file storage statistic was found without a preceding begin emipfs statistic. One or more file storage statistics have been skipped.

Condition Identifier: AVE\$MISSING_BEGIN_EMIPFS Code: AV 5455

Description: File storage statistics were encountered by ANALYZE_ACCOUNT_LOG without a preceding begin EMIPFS statistic. Possibly the EMIT_PERM_FILE_STATISTICS command is being used incorrectly.

User Action: Verify that the begin EMIPFS statistic has been activated to the log being processed.

--WARNING AV 5460-- A begin emipfs statistics was found when expecting an end emipfs statistic. Storage hours may be incorrect.

Condition Identifier: AVE\$MISSING_END_EMIPFS Code: AV 5460

Description: File storage statistics were encountered by ANALYZE_ACCOUNT_LOG in an unexpected order. Possibly the EMIT_PERM_FILE_STATISTICS command is being used incorrectly.

User Action: Verify that the desired file storage statistics have been activated to the log being processed.

--ERROR AV 5505-- Only accounting or family or system administrators may analyze usage for alternate families.

Condition Identifier: AVE\$NOT_YOUR_FAMILY Code: AV 5505

Description: A family name other than the current execution family name was specified for the FAMILY_NAME parameter on an ANALYZE_RESOURCE_USAGE subcommand.

User Action: Specify the current execution family name on the FAMILY_NAME parameter.

--ERROR AV 5510-- Only accounting or family or system administrators may analyze usage for alternate users.

Condition Identifier: AVE\$NOT_YOUR_USER_NAME Code: AV 5510

Description: A user name other than the current execution user name was specified for the USER_NAME parameter on an ANALYZE_RESOURCE_USAGE subcommand.

User Action: Specify the current execution user name for the USER_NAME parameter.

--ERROR AV 5515-- Version {text} is an old or unknown version of the resource usage file.

Condition Identifier: AVE\$INCORRECT_VERSION Code: AV 5515

Description: The resource usage file referenced is unreadable by the current version of the Accounting Analysis System.

User Action: Use the supplied Resource Usage File conversion routines to convert the Resource usage file to the new format.

--ERROR AV 5520-- The resource usage file {text} is empty.

Condition Identifier: AVE\$RUF_EMPTY Code: AV 5520

Description: The resource usage file referenced by ANALYZE_RESOURCE_USAGE is empty.

User Action: Use a resource usage file that contains information.

--ERROR AV 5525-- Terminating resource usage to an already existing file is not allowed.

Condition Identifier: AVE\$MUST_BE_NEW_FILE Code: AV 5525

Description: The result Resource Usage file referenced by TERMINATE_RESOURCE_USAGE already exists.

User Action: Reference a new file for the result of TERMINATE_RESOURCE_USAGE.

--ERROR AV 5530-- The resource usage file {text} must be converted to the new format by using the CONVERT_RESOURCE_USAGE command.

Condition Identifier: AVE\$RUF_NEEDS_CONVERSION Code: AV 5530

Description: The specified resource usage file is not in the correct format.

User Action: Convert the resource usage file to the correct format by executing the CONVERT_RESOURCE_USAGE command.

--ERROR AV 5605-- AVM\$PROCESS_SIGNED_REAL_VALUE can only be used to evaluate a SIGNED_REAL type.

Condition Identifier: AVE\$NOT_SIGNED_REAL_VALUE Code: AV 5605

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5610-- Input value given for parameter is not a signed real number.

Condition Identifier: AVE\$INPUT_NOT_SIGNED_REAL Code: AV 5610

Description: A value was specified for a signed real parameter that is not a signed real number.

User Action: Specify a signed real value for the parameter.

--ERROR AV 5705-- {text} is the resource identifier of an existing resource price.

Condition Identifier: AVE\$RESOURCE_ID_CONFLICT Code: AV 5705

Description: Resource identifiers on the resource prices file must be unique.

User Action: Use a resource identifier that is different from all other resource identifiers on the resource prices file.

--ERROR AV 5710-- {text} is the resource name of an existing resource prices.

Condition Identifier: AVE\$RESOURCE_NAME_CONFLICT Code: AV 5710

Description: Resource names on the resource prices file must be unique.

User Action: Use a resource name that is different than all other resource names on the resource prices file.

--ERROR AV 5715-- {text} is not a resource identifier of any resource price.

Condition Identifier: AVE\$RESOURCE_ID_NOT_FOUND Code: AV 5715

Description: The resource identifier entered on an ANALYZE_RESOURCE_PRICES subcommand does not exist on the Resource Prices file.

User Action: Use a resource identifier that exists on the Resource Prices file.

--ERROR AV 5720-- {text} is not a resource name of any resource price.

Condition Identifier: AVE\$RESOURCE_NAME_NOT_FOUND Code: AV 5720

Description: The resource name entered on an ANALYZE_RESOURCE_PRICES subcommand does not exist on the Resource Prices file.

User Action: Use a resource name that exists on the Resource Prices file.

--ERROR AV 5723-- The resource price file {text} must be converted to the new format by using the ADMINISTER_RESOURCE_PRICES utility.

Condition Identifier: AVE\$RPF_NEEDS_CONVERSION Code: AV 5723

Description: The specified resource price file is not in the correct format.

User Action: Convert the resource price file to the correct format by using the ADMINISTER_RESOURCE_PRICES utility.

--ERROR AV 5725-- The system resource prices file {text} does not have any resource prices.

Condition Identifier: AVE\$SYSTEM_RPF_EMPTY Code: AV 5725

Description: The system Resource Prices file has no pricing entries on it.

User Action: Enter resource prices using the ADMINISTER_RESOURCE_PRICES utility.

--ERROR AV 5730-- The system resource prices file {text} either does not exist or you are not allowed access to it.

Condition Identifier: AVE\$SYSTEM_RPF_NOT_FOUND Code: AV 5730

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a Programming System Report (PSR).

--ERROR AV 5735-- Only an accounting or system administrator may access {text}.

Condition Identifier: AVE\$SYS_ADMIN_REQ_FOR_ACCESS Code: AV 5735

Description: The listed file is a secure file that only the system administrator may access.

User Action: None.

--ERROR AV 5740-- Only an accounting or system administrator may update {text}.

Condition Identifier: AVE\$SYS_ADMIN_REQ_FOR_UPDATE Code: AV 5740

Description: The listed file is a secure file that only the system administrator may update.

User Action: None.

--ERROR AV 5808-- {text} was called when the file was locked for read access.

Condition Identifier: AVE\$IFM_CONFLICTING_LOCK Code: AV 5808

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5816-- An internal error was detected in the index records for the indexed file.

Condition Identifier: AVE\$IFM_CORRUPTED_INDEX_RECORD Code: AV 5816

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5824-- An internal error prevented {text} from retrieving the value of the variable {text} from the sequence {text}.

Condition Identifier: AVE\$IFM_CORRUPTED_SEQUENCE Code: AV 5824

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5832-- The end of the file was encountered.

Condition Identifier: AVE\$IFM_END_OF_FILE Code: AV 5832

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5840-- {file} is not a indexed file.

Condition Identifier: AVE\$IFM_NOT_AN_INDEXED_FILE Code: AV 5840

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5848-- A record with the key {text} already exists.

Condition Identifier: AVE\$IFM_RECORD_ALREADY_EXISTS Code: AV 5848

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5856-- The file {text} is full.

Condition Identifier: AVE\$IFM_FILE_FULL Code: AV 5856

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5864-- A record with the key {text} could not be found.

Condition Identifier: AVE\$IFM_UNKNOWN_RECORD Code: AV 5864

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5872-- An unknown file identifier was passed on a call to the indexed file manager.

Condition Identifier: AVE\$IFM_UNKNOWN_INDEXED_FILE_ID Code: AV 5872

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--ERROR AV 5880-- The file must be locked for update access before calling {text}.

Condition Identifier: AVE\$IFM_UPDATE_LOCK_REQUIRED Code: AV 5880

Description: An internal error has occurred.

User Action: Provide a site analyst with appropriate information or submit a PSR.

--WARNING BC 1-- Virtual BASIC source errors.

Condition Identifier: BCE\$COMPILER_SOURCE_WARNING Code: BC 1

Description: There are errors in the source program but the program was able to compile.

User Action: Correct the source program and recompile.

--FATAL BC 2-- Virtual BASIC source errors.

Condition Identifier: BCE\$COMPILER_SOURCE_FATAL Code: BC 2

Description: There are errors in the source program.

User Action: Correct the source program and recompile.

--CATASTROPHIC BC 3-- Virtual BASIC source errors.

Condition Identifier: BCE\$COMPILER_SOURCE_CATASTROPHE Code: BC 3

Description: There are errors in the source program.

User Action: Correct the source program and recompile.

--FATAL BC 100-- The second letter alphabetically precedes the first.

Condition Identifier: BCE\$TYPE_OUT_OF_ORDER Code: BC 100

Description: In the type declaration statement you have specified a list of letter and letter ranges which is out of order.

User Action: Correct the type declaration statement.

--CATASTROPHIC BC 410-- Compiler failure -- DATA statement has empty body.

Condition Identifier: BCE\$DATA_BUG Code: BC 410

Description: There is a bug in the compiler.

User Action: Contact your site analyst for further information.

--FATAL BC 525-- THEN required to follow ELSEIF at line {text}, column {text}.

Condition Identifier: BCE\$THEN_REQUIRED Code: BC 525

Description: The keyword THEN must follow the keyword ELSEIF.

User Action: Supply the appropriate keyword THEN. Check for extraneous characters on either side of the keyword THEN.

--FATAL BC 4025-- The INPUT file {text} is not available.

Condition Identifier: BCE\$NO_RES_INPUT Code: BC 4025

Description: The input file specified for the RESEQUENCE utility is not found.

User Action: Check that the correct file was specified.

--FATAL BC 4030-- The INPUT file {text} is not available for READ access.

Condition Identifier: BCE\$RES_INPUT_UNREADABLE Code: BC 4030

Description: The input file specified for the RESEQUENCE utility is not available for READ access.

User Action: Check that the correct file was specified. Use the SET_FILE_ATTRIBUTES command to change the input file ACCESS_MODE to READ. To display information about a file use the DISPLAY_FILE_ATTRIBUTES command.

--FATAL BC 4035-- The INPUT file {text} specified for RESEQUENCE is empty.

Condition Identifier: BCE\$RES_INPUT_EMPTY Code: BC 4035

Description: The input file specified for the RESEQUENCE utility is empty.

User Action: Check that the correct file was specified.

--FATAL BC 4040-- The INPUT file {text} specified for RESEQUENCE cannot be read as a sequential file.

Condition Identifier: BCE\$RES_INPUT_NOT_SEQ Code: BC 4040

Description: The input file specified for the RESEQUENCE utility is defined with attributes that prevent it from being read as a sequential file.

User Action: Check that the correct file was specified. Use the SET_FILE_ATTRIBUTES command to change the FILE_ORGANIZATION to SEQUENTIAL. To display information about a file use the DISPLAY_FILE_ATTRIBUTES command.

--FATAL BC 4055-- Empty OUTPUT file {text} specified for RESEQUENCE must have APPEND access.

Condition Identifier: BCE\$CANNOT_APPEND_RES_OUT Code: BC 4055

Description: The output file specified for the RESEQUENCE utility is empty and is not available for append access.

User Action: Check that the correct file was specified. Use the SET_FILE_ATTRIBUTES command to change the output file ACCESS_MODE to APPEND. To display information about a file use the DISPLAY_FILE_ATTRIBUTES command.

--FATAL BC 4065-- OUTPUT file {text} specified for RESEQUENCE cannot be written as a sequential file.

Condition Identifier: BCE\$RES_OUT_NOT_SEQ Code: BC 4065

Description: The output file specified for the RESEQUENCE utility is defined with attributes that prevent it from being written as a sequential file.

User Action: Check that the correct file was specified. Use the SET_FILE_ATTRIBUTES command to change the output file FILE_ORGANIZATION to SEQUENTIAL. To display information about a file use the DISPLAY_FILE_ATTRIBUTES command.

--FATAL BC 4070-- Resequenced line {text} exceeds the maximum record length allowed on file {text}.

Condition Identifier: BCE\$RES_OUT_RECLLEN_EXCEEDED Code: BC 4070

Description: The output file specified for the RESEQUENCE utility is a preexisting non-empty file with a record type of fixed length and a line of the resequenced text exceeds the length defined for the files records.

User Action: Check that the correct file was specified. Use the SET_FILE_ATTRIBUTES command to change the output file RECORD_TYPE to VARIABLE. The default record type is VARIABLE. To display information about a file use the DISPLAY_FILE_ATTRIBUTES command.

--FATAL BC 4120-- A new label generated from RESEQUENCE parameters exceeds 999,999, the maximum label value.

Condition Identifier: BCE\$NEW_LABEL_TOO_LARGE Code: BC 4120

Description: The result of applying the new base and new increment to the input file lead to a new label with a number value greater than 999,999 (that is, if $(NLL - 1 * NI) + NB$ is greater than 999,999, where NLL is the number of labeled lines, NI is the new increment, and NB is the new base).

User Action: Supply a valid new label and modify all label references that are affected by the change.

--FATAL BC 4125-- The old label defined at line {text} exceeds 999,999, the maximum label value.

Condition Identifier: BCE\$OLD_LABEL_TOO_LARGE Code: BC 4125

Description: The input file specified contains a reference of a label the value of which exceeds the maximum legal label value.

User Action: Supply a valid label and modify all label references that are affected by the change.

--FATAL BC 4130-- The label referenced at line {text}, column {text} is not defined.

Condition Identifier: BCE\$OLD_LABEL_UNDEFINED Code: BC 4130

Description: The input file contains a reference to a label that is not defined in the containing routine.

User Action: Make sure you referenced the correct label. Provide the appropriate line with the label (integer).

--FATAL BC 4135-- The label definition at line {text} of the RESEQUENCE input file is not greater than the preceding label definition.

Condition Identifier: BCE\$LABEL_OUT_OF_ORDER Code: BC 4135

Description: The input file contains a label definition the value of which is not greater than that of the preceding label defined in the same routine.

User Action: Relabel the line, and others as needed, so that the labels in the external routine form an increasing sequence.

--WARNING BC 4150-- No labels were found in the INPUT file {text}.

Condition Identifier: BCE\$NO_LABEL_FOUND Code: BC 4150

Description: No labels are found in the input file specified for the RESEQUENCE utility.

User Action: Check that the correct input file was specified. If it was, this is a BASIC program that has no need for resequencing.

--WARNING BC 4160-- Resequenced line {text} exceeds the maximum length for a BASIC source program.

Condition Identifier: BCE\$RES_LINE_TOO_LONG Code: BC 4160

Description: The number of characters in the resequenced file exceeds the NOS/VE BASIC maximum line length of 255 (integer).

User Action: Split the source line into two or more valid lines.

--FATAL BC 5003-- ERR = {text}, ERL = {text} in module {text}: Divide fault.

Condition Identifier: BCE\$DIVIDE_FAULT Code: BC 5003

Description: Division by zero is a mathematically undefined operation.

User Action: Correct the expression.

--FATAL BC 5006-- The size of the array passed to BCPDAUA must be from 1 to 63 words.

Condition Identifier: BCE\$APPLIC_ARRAY Code: BC 5006

Description: The array passed to the BCPDAUA subroutine must be a single-dimension array of 1 to 63 integers. The array you passed to BCPDAUA is not within the range 1 to 63.

User Action: Correct the size of the array you are passing to the BCPDAUA subroutine.

--FATAL BC 5010-- ERR = {text}, ERL = {text} in module {text}: Storage required for array exceeds memory segment.

Condition Identifier: BCE\$LEVIATHAN Code: BC 5010

Description: The maximum number of elements for an array depends on the maximum contiguous storage allowed for your installation account.

User Action: See your site administrator for specific information.

--FATAL BC 5102-- ERR = {text}, ERL = {text} in module {text}: Array subscript upper bound cannot be less than lower bound.

Condition Identifier: BCE\$LB_GT_UB Code: BC 5102

Description: The lower bound of an array must be less than or equal to the upper bound of the array.

User Action: Correct the bounds of the array.

--FATAL BC 5103-- ERR = {text}, ERL = {text} in module {text}: Magnitude of array bound cannot exceed $2^{31}-1$.

Condition Identifier: BCE\$BOUNDS_MAGNITUDE Code: BC 5103

Description: The magnitude of an array bound exceeds the maximum array bound size of $(2^{31}-1)$

User Action: Replace the invalid array bound with a valid array bound.

--FATAL BC 5270-- ERR = {text}, ERL = {text} in module {text}: Logarithm of zero or negative number.

Condition Identifier: BCE\$LOG_OF_NON_POSITIVE Code: BC 5270

Description: Logarithm is the exponent that indicates the power to which a number is raised. A logarithm cannot be defined for zero or a negative number.

User Action: Correct the logarithm.

--FATAL BC 5310-- ERR = {text}, ERL = {text} in module {text}: COS argument cannot equal or exceed 2^{47} .

Condition Identifier: BCE\$COS_ARG_ERROR Code: BC 5310

Description: The value of the argument in the intrinsic function COS must be less than 2^{47} .

User Action: Correct the value of the argument so it is less than 2^{47} .

--FATAL BC 5340-- ERR = {text}, ERL = {text} in module {text}: Absolute value of COSH argument cannot equal or exceed $4095 \cdot \text{LOG}(2)$.

Condition Identifier: BCE\$COSH_ARG_ERROR Code: BC 5340

Description: The absolute value of the argument in the intrinsic function COSH must be less than $4095 \cdot \text{LOG}(2)$.

User Action: Correct the value of the argument so it is less than $4095 \cdot \text{LOG}(2)$.

--FATAL BC 5360-- ERR = {text}, ERL = {text} in module {text}: Absolute value of EXP argument cannot equal or exceed $4095 \cdot \text{LOG}(2)$.

Condition Identifier: BCE\$EXP_ARG_ERROR Code: BC 5360

Description: The absolute value of the argument in the intrinsic function EXP must be less than $4095 \cdot \text{LOG}(2)$.

User Action: Correct the value of the argument so it is less than $4095 \cdot \text{LOG}(2)$.

--FATAL BC 5400-- ERR = {text}, ERL = {text} in module {text}: Argument of CHR\$ out of range.

Condition Identifier: BCE\$CHR_ARG_ERROR Code: BC 5400

Description: The value of the argument in the CHR\$ function must range between 0 and 255.

User Action: Supply a valid ASCII code between 0 and 255.

--FATAL BC 5405-- ERR = {text}, ERL = {text} in module {text}: Null argument for ASC.

Condition Identifier: BCE\$ASC_ARG_ERROR Code: BC 5405

Description: The ASC function was passed the value of string, which was a null string.

User Action: Verify that a valid string expression was supplied.

--FATAL BC 5420-- ERR = {text}, ERL = {text} in module {text}: CVI, CVS, or CVD function argument length must be 8.

Condition Identifier: BCE\$CVX_ARG_LEN_NOT_8 Code: BC 5420

Description: The CVI, CVS, or CVD function arguments must have a length of 8 bytes.

User Action: Correct the value of the argument so its length is

--FATAL BC 5430-- ERR = {text}, ERL = {text} in module {text}: Substring length negative in LEFT\$ or RIGHT\$ reference.

Condition Identifier: BCE\$LEFT\$_RIGHT\$_LEN_NEG Code: BC 5430

Description: The value of length must be positive in the LEFT\$ or RIGHT\$ function.

User Action: Correct the value of length so that it is positive.

--FATAL BC 5435-- ERR = {text}, ERL = {text} in module {text}: First character position not positive in MID\$ reference.

Condition Identifier: BCE\$MID\$_NON_POS_FCP Code: BC 5435

Description: In the MID\$ reference the value of position must be positive.

User Action: Change the reference to positive.

--FATAL BC 5440-- ERR = {text}, ERL = {text} in module {text}: Length negative in MID\$ reference.

Condition Identifier: BCE\$MID\$_LEN_NEG Code: BC 5440

Description: In the MID\$ reference the value of length must be positive.

User Action: Correct the value of length so that it is positive.

--FATAL BC 5441-- ERR = {text}, ERL = {text} in module {text}: Insertion beyond target string length in MID\$ reference.

Condition Identifier: BCE\$MID\$_INSERT_BEYOND_TARGET Code: BC 5441

Description: MID\$ reference tried to insert the string beyond the length of the specified string.

User Action: Correct MID\$ reference.

--FATAL BC 5445-- ERR = {text}, ERL = {text} in module {text}: INSTR index is out of range.

Condition Identifier: BCE\$INSTR_INDEX_ERROR Code: BC 5445

Description: The INSTR string index value must not be less than 1 or greater than the maximum string length.

User Action: Correct the INSTR string so it is within the allowable range.

--FATAL BC 5450-- ERR = {text}, ERL = {text} in module {text}: Improper value for DATE\$.

Condition Identifier: BCE\$DATE\$_ERROR Code: BC 5450

Description: DATE\$ is a 10-character string variable. The value of DATE\$ has the format: mm-dd-yyyy.

User Action: Correct the value for DATE\$.

--FATAL BC 5500-- ERR = {text}, ERL = {text} in module {text}: File is not available to program.

Condition Identifier: BCE\$FILE_UNAVAILABLE Code: BC 5500

Description: The file named on an OPEN statement was not found.

User Action: Check that the correct name was furnished. If the file exists, check that you have the required access parameters.

--FATAL BC 5501-- ERR = {text}, ERL = {text} in module {text}: Filename is not a valid NOS/VE local file name.

Condition Identifier: BCE\$INVALID_FILE_NAME Code: BC 5501

Description: A valid NOS/VE local file name is 1 to 31 alphabetic characters, digits, and/or four special characters, beginning with a nondigit.

User Action: Supply a valid NOS/VE local file name.

--FATAL BC 5510-- ERR = {text}, ERL = {text} in module {text}: Channel number not in range.

Condition Identifier: BCE\$CHANNEL_RANGE_ERROR Code: BC 5510

Description: Channel number must range between 0 and 99.

User Action: Supply a valid channel number between 0 and 99

--FATAL BC 5511-- ERR = {text}, ERL = {text} in module {text}: Channel is not open.

Condition Identifier: BCE\$CHANNEL_INACTIVE Code: BC 5511

Description: Check the channel value. If it is the one intended, supply an OPEN statement. The OPEN statement opens a channel through which a BASIC program can access a NOS/VE file.

User Action: Not applicable.

--FATAL BC 5512-- ERR = {text}, ERL = {text} in module {text}: Channel is already open.

Condition Identifier: BCE\$CHANNEL_ACTIVE Code: BC 5512

Description: Channel has already been opened by the OPEN statement.

User Action: Check that the channel value was the one intended.

--FATAL BC 5513-- ERR = {text}, ERL = {text} in module {text}: Channel zero cannot be specified on OPEN or CLOSE statement.

Condition Identifier: BCE\$CHANNEL_ZERO_ERROR Code: BC 5513

Description: Channel number for the OPEN and CLOSE statements must range between 1 and 99.

User Action: Supply a valid channel number between 1 and 99.

--FATAL BC 5514-- ERR = {text}, ERL = {text} in module {text}: Channel zero cannot be specified for the LOC function.

Condition Identifier: BCE\$LOC_ZERO Code: BC 5514

Description: The channel number for the LOC function must range between 1 and 99.

User Action: Correct the channel number so that it is between 1 and 99.

--FATAL BC 5525-- ERR = {text}, ERL = {text} in module {text}: Program has insufficient access to file.

Condition Identifier: BCE\$INSUFFICIENT_ACCESS Code: BC 5525

Description: The program cannot access the file due to a discrepancy in the access mode parameter.

User Action: Change the ACCESS_MODE parameter on the ATTACH_FILE command. Check the access permissions available to your file.

--FATAL BC 5540-- ERR = {text}, ERL = {text} in module {text}: Local file limit has been reached.

Condition Identifier: BCE\$LOCAL_FILE_LIMIT Code: BC 5540

Description: The local file limit is approximately 70. No more than 70 files can be used in a program.

User Action: Reduce the number of local files to no more than 70.

--FATAL BC 5600-- ERR = {text}, ERL = {text} in module {text}: Length greater than record length of file.

Condition Identifier: BCE\$LENGTH_GT_RECLEN Code: BC 5600

Description: The specified record length of an OPEN statement of a preexisting sequential file cannot exceed the maximum record length attribute of the file.

User Action: Reduce the record length.

--FATAL BC 5605-- ERR = {text}, ERL = {text} in module {text}: Record length must be greater than zero.

Condition Identifier: BCE\$INVALID_RECLEN Code: BC 5605

Description: The specified record length must be greater than zero and not exceed the maximum record length allowed.

User Action: Correct the record length so that it is within a permissible range.

--FATAL BC 5610-- ERR = {text}, ERL = {text} in module {text}: Invalid io-mode specified on OPEN statement.

Condition Identifier: BCE\$INVALID_IO_MODE Code: BC 5610

Description: The string supplied on the OPEN statement does not begin with I, O, or R.

User Action: Correct the OPEN statement and supply a valid I/O mode string.

--FATAL BC 5615-- ERR = {text}, ERL = {text} in module {text}: File cannot be opened in accordance with specified io-mode.

Condition Identifier: BCE\$INVALID_ACCESS Code: BC 5615

Description: The NOS/VE file organization and the BASIC I/O mode are not compatible.

User Action: Correct the I/O mode on the OPEN statement. (Valid options: INPUT, OUTPUT, APPEND, and RANDOM.) Be sure that you have the appropriate access permissions to the file.

--FATAL BC 5618-- ERR = {text}, ERL = {text} in module {text}: File created as random cannot be opened as sequential.

Condition Identifier: BCE\$FILE_NOT_SEQUENTIAL Code: BC 5618

Description: A file that is created as random must be opened as RANDOM. A file that is created as sequential must be opened as SEQUENTIAL.

User Action: OPEN statement must specify an I/O mode of RANDOM. (NOS/VE file organization of BYTE_ADDRESSABLE)

--FATAL BC 5620-- ERR = {text}, ERL = {text} in module {text}: Indexed sequential file cannot be OPENed by BASIC.

Condition Identifier: BCE\$INDEXED_SEQUENTIAL_FILE Code: BC 5620

Description: BASIC does not support indexed sequential files.

User Action: Check that the file named on the OPEN statement is the one intended.

--FATAL BC 5621-- ERR = {text}, ERL = {text} in module {text}: Random file must have BYTE ADDRESSABLE file organization attribute.

Condition Identifier: BCE\$FILE_NOT_RANDOM Code: BC 5621

Description: The file organization attribute must be set to BYTE_ADDRESSABLE for a random file.

User Action: To change FILE_ORGANIZATION for a random file, use the SET_FILE_ATTRIBUTES command with FILE_ORGANIZATION = BYTE_ADDRESSABLE. To display information about a file, use the DISPLAY_FILE_ATTRIBUTE command.

--FATAL BC 5630-- ERR = {text}, ERL = {text} in module {text}: File to be opened for INPUT is not attached.

Condition Identifier: BCE\$INPUT_ETHER Code: BC 5630

Description: The file to be opened has not been attached by the SCL ATTACH_FILE command.

User Action: Attach file before attempting to open.

--FATAL BC 5631-- ERR = {text}, ERL = {text} in module {text}: File to be opened for INPUT is empty.

Condition Identifier: BCE\$INPUT_EMPTY Code: BC 5631

Description: The file specified on the OPEN statement with an access mode of INPUT is empty.

User Action: Verify the contents of the input file. Check that the correct file was named.

--FATAL BC 5650-- ERR = {text}, ERL = {text} in module {text}: Cannot GET from a file opened as sequential.

Condition Identifier: BCE\$GET_FROM_SEQUENTIAL Code: BC 5650

Description: The GET statement only permits access when an I/O mode of RANDOM is specified.

User Action: OPEN statement must specify an I/O mode of RANDOM (NOS/VE file organization of BYTE_ADDRESSABLE).

--FATAL BC 5671-- ERR = {text}, ERL = {text} in module {text}: No data available in random buffer.

Condition Identifier: BCE\$BUFFER_EMPTY Code: BC 5671

Description: No data has been read to the random buffer.

User Action: Verify that the data has been read into the random buffer before accessing.

--FATAL BC 5672-- ERR = {text}, ERL = {text} in module {text}: Too few data in record.

Condition Identifier: BCE\$RECORD_INSUFFICIENT Code: BC 5672

Description: You are trying to read a record that contains not enough data.

User Action: Check the record you are trying to read.

--FATAL BC 5673-- ERR = {text}, ERL = {text} in module {text}: Too many data in record.

Condition Identifier: BCE\$RECORD_EXCESSIVE Code: BC 5673

Description: You are trying to read a record that contains too much data.

User Action: Check the record you are trying to read.

--FATAL BC 5675-- ERR = {text}, ERL = {text} in module {text}: End of file encountered.

Condition Identifier: BCE\$EOF_ENCOUNTERED Code: BC 5675

Description: The end of file was encountered.

User Action: Ensure that the file contains all required data.

--FATAL BC 5680-- ERR = {text}, ERL = {text} in module {text}: Syntax error in input reply.

Condition Identifier: BCE\$INPUT_SYNTAX_ERROR Code: BC 5680

Description: The items of an input reply are ill-formed or not separated by commas.

User Action: Check the input file.

--FATAL BC 5690-- ERR = {text}, ERL = {text} in module {text}: Datum format error.

Condition Identifier: BCE\$DATUM_FORMAT_ERROR Code: BC 5690

Description: Erroneous data has been encountered in the format of the INPUT statement.

User Action: Check for erroneous data. (ex. missing quotes, too few or too many values, misspelled values)

--FATAL BC 5691-- ERR = {text}, ERL = {text} in module {text}: Datum type not conformable to variable.

Condition Identifier: BCE\$DATUM_TYPE_ERROR Code: BC 5691

Description: The value of an expression falls outside of the range allowed for variables of the corresponding data type.

User Action: Supply a value within the range allowed for typed variables.

--FATAL BC 5695-- ERR = {text}, ERL = {text} in module {text}: Invalid format string.

Condition Identifier: BCE\$INVALID_FORMAT Code: BC 5695

Description: The format string is inappropriate for the data type specified on the corresponding print list item.

User Action: Correct the format string so that it matches the data type specified.

--FATAL BC 5697-- ERR = {text}, ERL = {text} in module {text}: Format item inappropriate for PRINT datum.

Condition Identifier: BCE\$IMPROPER_FORMAT Code: BC 5697

Description: The current datum formatting field of a PRINT USING statement is of the wrong type for the current item of the print list.

User Action: Check that the types of print list items correspond to the formatting fields.

--FATAL BC 5711-- ERR = {text}, ERL = {text} in module {text}: Buffer length exceeded on input from file.

Condition Identifier: BCE\$INPUT_RECLLEN_EXCEEDED Code: BC 5711

Description: Attempted to read the input file and received a length that exceeds the length specified when the file was opened.

User Action: Increase the length specified on the OPEN statement.

--FATAL BC 5720-- ERR = {text}, ERL = {text} in module {text}: FIELD specifications exceed record length.

Condition Identifier: BCE\$FIELD_OVERFLOW Code: BC 5720

Description: In the FIELD statement, the number of bytes specified exceeds the length of the buffer.

User Action: Reduce the number of bytes specified.

--FATAL BC 5734-- ERR = {text}, ERL = {text} in module {text}: Width of fixed-length record file cannot be changed.

Condition Identifier: BCE\$FIXED_WIDTH Code: BC 5734

Description: The width of a fixed-length record cannot be changed.

User Action: The WIDTH statement is not valid with fixed-length records.

--FATAL BC 5735-- ERR = {text}, ERL = {text} in module {text}: A file other than OUTPUT or PRINT cannot be named on a WIDTH statement.

Condition Identifier: BCE\$IMPROPER_WIDTH_FNAME Code: BC 5735

Description: Files OUTPUT and PRINT are the only file names that can be specified on the WIDTH statement.

User Action: Supply a valid file name.

--FATAL BC 5750-- ERR = {text}, ERL = {text} in module {text}: Device failure.

Condition Identifier: BCE\$DEVICE_FAILURE Code: BC 5750

Description: Internal problem with the NOS/VE operating system.

User Action: Follow the procedure established at your site for reporting software problems.

--FATAL BC 6010-- ERR = {text}, ERL = {text} in module {text}: Unresolved error from called routine.

Condition Identifier: BCE\$ERROR_FROM_CALL Code: BC 6010

Description: When returning from a called routine, an error resulted because of an uncleared error.

User Action: Correct the error or use the RESUME statement.

--FATAL BC 7000-- ERR = {text}, ERL = {text} in module {text}: Unrecognized error.

Condition Identifier: BCE\$ERROR_INCOGNITA Code: BC 7000

Description: The ERROR statement did not recognize the error number as a NOS/VE BASIC runtime error.

User Action: Not applicable.

--CATASTROPHIC BC 7002-- ERR = {text}, ERL = {text} in module {text}: Additional error encountered while executing error handling code. Execution terminated.

Condition Identifier: BCE\$COMPOUND_ERRORS Code: BC 7002

Description: Concurrent errors in an environment cannot be handled.

User Action: Check the error file and make the requested correction.

--CATASTROPHIC BC 7003-- ERR = {text}, ERL = {text} in module {text}: Hardware arithmetic error raised in non-BASIC routine cannot be handled by BASIC.

Condition Identifier: BCE\$XENOPHOBIA Code: BC 7003

Description: BASIC error handling has been invoked by an error in a non-BASIC routine.

User Action: A routine that is not compiled by BASIC lacks BASIC's error handling information. Try debugging the foreign routine separately.

--CATASTROPHIC BC 7004-- ERR = {text}, ERL = {text} in module {text}: Unanticipated response from the system interface.

Condition Identifier: BCE\$SYSTEM_ANOMALY Code: BC 7004

Description: Undocumented response from system interface.

User Action: Follow the procedure established at your site for reporting software problems.

--CATASTROPHIC BC 7005-- ERR = {text}, ERL = {text} in module {text}: Line address tables have been removed from module; unable to RESUME.

Condition Identifier: BCE\$MISSING_LINE_TABLE Code: BC 7005

Description: The line address tables have been removed. Program is unable to RESUME execution in the current environment.

User Action: The object code utility has been used to remove line address tables. Recompile your program to generate new line address tables.

--CATASTROPHIC CB 11-- SY and OC both used.

Condition Identifier: CBE\$SY_USED_WITH_OC Code: CB 11

Description: Combination of 'SY' and 'OC' values for parameter DEBUG_AIDS not allowed.

User Action: Specify either 'OC' to produce object code and have compilation diagnostics issued or 'SY' to perform syntax checking only. You cannot specify both.

Further Information: See the description of the DEBUG_AIDS parameter of the COB command in the COBOL Usage manual.

--CATASTROPHIC CB 16-- " or ' required.

Condition Identifier: CBE\$VALUE_FOR_LC_INCORRECT Code: CB 16

Description: Parameter LC allows only apostrophe or quotation mark as value.

User Action: Specify either the apostrophe or quotation mark to delimit nonnumeric literals. No other options are allowed.

Further Information: See the description of the LITERAL_CHARACTER parameter of the COBOL command in the COBOL Usage manual.

--CATASTROPHIC CB 21-- I or W or F required for ANSI.

Condition Identifier: CBE\$KEYWORD_VALUE_REQUIRED Code: CB 21

Description: ANSI value for parameter STANDARD_DIAGNOSTICS requires second value T, W, or F.

User Action: Specify the severity level of ANSI diagnostics to be indicated. The format is STANDARD_DIAGNOSTICS=(ANSI,s).

Further Information: See the description of the STANDARD_DIAGNOSTICS parameter of the COBOL command in the COBOL Usage manual.

--ERROR CB 50-- Fatal compilation errors.

Condition Identifier: CBE\$COMPILATION_ERRORS Code: CB 50

Description: "--ERROR-- Fatal compilation errors." The compiler found errors in the input source text that prevented successful compilation. Messages describing the errors were written to the file specified by the ERRORS (E) command parameter.

User Action: List the file specified on the ERRORS (E) file. The default ERRORS file is the file specified by the LIST (L) parameter if one was specified; otherwise, the default is \$ERRORS.

--CATASTROPHIC CB 55-- Empty source file.

Condition Identifier: CBE\$NO_SOURCE Code: CB 55

Description: The file named as containing the source program does not exist or is empty or has an incorrect file attribute.

User Action: Check the INPUT parameter on the COBOL command to see if you have named the source file correctly. If so, make sure that the file is positioned at its beginning, by rewinding if necessary. Also ensure that the FILE_CONTENTS file attribute of the source file is not LIST.

Further Information: See the description of the INPUT parameter of the COBOL command in the COBOL Usage manual.

--CATASTROPHIC CB 60-- Compiler aborted due to {text}.

Condition Identifier: CBE\$COMPILER_ABORTED Code: CB 60

Description: The compiler was aborted for the indicated reason.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--WARNING CB 65-- The binary file specified was an existing file of UNKNOWN contents. To use the data generated to that file, it is necessary to issue the following SCL command : CHAFA LFN FC=OBJECT FS=DATA. LFN is the name specified for the BINARY parameter.

Condition Identifier: CBE\$BAD_FILE_CONTENTS Code: CB 65

Description: The FILE_CONTENTS attribute value for the file specified by the BINARY_OBJECT (B) parameter is UNKNOWN.

User Action: Check that the correct file was specified on the BINARY_OBJECT (B) parameter. Change the FILE_CONTENTS and FILE_STRUCTURE attribute values using this command:

```
change_file_attribute, file=name_of_binary_file, ..
file_contents=object, file_structure=data
```

Further Information: For a description of the FILE_CONTENTS and FILE_STRUCTURE attributes, see the SCL System Interface manual.

For the CHANGE_FILE_ATTRIBUTE command description, see the SCL Quick Reference manual.

--ERROR CB 70-- The FILE_CONTENTS attribute for the BINARY_OBJECT file must be either UNKNOWN or OBJECT.

Condition Identifier: CBE\$BAD_OBJECT_FILE Code: CB 70

Description: The FILE_CONTENTS attribute value for the file specified by the BINARY_OBJECT (B) parameter is not UNKNOWN or OBJECT.

User Action: Check whether the correct file was specified on the BINARY_OBJECT (B) parameter. If appropriate, change the FILE_CONTENTS attribute value to OBJECT using this command:

```
change_file_attribute, file=name_of_binary_file, ..
file_contents=object
```

Further Information: For a description of the FILE_CONTENTS attribute, see the SCL System Interface manual.

For the CHANGE_FILE_ATTRIBUTE command description, see the SCL Quick Reference manual.

--ERROR CB 75-- The input file source text cannot be read because the FILE_CONTENTS attribute is not LEGIBLE or UNKNOWN.

Condition Identifier: CBE\$BAD_INPUT_FILE Code: CB 75

Description: The FILE_CONTENTS attribute value for the file specified by the INPUT (I) parameter is not UNKNOWN or LEGIBLE.

User Action: Check whether the correct file was specified on the INPUT (I) parameter. If appropriate, change the FILE_CONTENTS attribute value using one of these commands:

```
change_file_attribute, file=name_of_input_file, ..
file_contents=unknown
```

```
change_file_attribute, file=name_of_input_file, ..
file_contents=legible
```

Further Information: For a description of the FILE_CONTENTS attribute, see the SCL System Interface manual.

For the CHANGE_FILE_ATTRIBUTE command description, see the SCL Quick Reference manual.

--FATAL CB 9200-- COBOL internal error: run time diagnostic number too low.

Condition Identifier: CBE\$RUN_TIME_ERRORS Code: CB 9200

Description: During the execution of COBOL object code or COBOL generated code, a COBOL runtime routine is trying to issue a diagnostic with a diagnostic number that is too low.

User Action: Call the site analyst.

--FATAL CB 9205-- At line {text}, at end detected on ACCEPT from {file}.

Condition Identifier: CBE\$AT_END_ON_ACCEPT Code: CB 9205

Description: An at-end condition (end of information) occurred during execution of an ACCEPT statement. In batch, the run will be terminated. If an interactive user elects to continue, spaces will be returned.

User Action: Correct program logic to detect and process the last file record. Make sure that the file is properly positioned.

--FATAL CB 9210-- At line {text}, attempt to open the locked file {file}.

Condition Identifier: CBE\$OPEN_LOCKED_FILE Code: CB 9210

Description: The named file was previously closed by a CLOSE WITH LOCK statement in this run unit, and the current OPEN statement references it.

User Action: Remove the WITH LOCK phrase of the CLOSE statement, or change the OPEN statement.

Further Information: See the discussion of the CLOSE statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9215-- At line {text}, attempt to open file {file}, which has same name as an open file.

Condition Identifier: CB\$SAME_FILE_OPEN Code: CB 9215

Description: An OPEN statement executed on the named file, but the file is already open in this run unit. Perhaps the OPEN statement is associated with another COBOL file-name, and an OPEN statement for this file-name was previously issued.

User Action: Change program to reference correct file.

--FATAL CB 9220-- At line {text}, CALL of program {text} was not successful and no OVERFLOW phrase was specified. The loader status follows.

Condition Identifier: CB\$CALL_EXCEPTION_NO_OPTION Code: CB 9220

Description: A CALL statement referencing the named program failed due to an error status return from the loader. The loader error status is also shown. The most likely cause is that the program was not loaded with the run unit. Other possible causes are that the program-name is incorrect or that the library containing the program is not part of the local set.

User Action: To inspect contents of library, use the DISPLAY_OBJECT_LIBRARY command. To list global library set, use the DISPLAY_PROGRAM_ATTRIBUTE command. To list local library set, use the EXECUTE_TASK command.

--FATAL CB 9230-- At line {text}, {text} executed on file {file}, which is open.

Condition Identifier: CB\$OPEN_OR_DF_ON_OPEN_FILE Code: CB 9230

Description: A DELETE FILE or OPEN statement was executed on the named file, and the file was open when the statement executed. If continuation is elected, the file will be closed as if a CLOSE statement with no options were executed and the normal DELETE FILE or OPEN process will continue.

User Action: Close the named file.

--WARNING CB 9235-- At line {text}, {text} on file {file}, which has sequential access and no valid READ preceded the operation.

Condition Identifier: CB\$DEL_OR_REW_WITH_NO_READ Code: CB 9235

Description: The immediately preceding operation on the file was not a valid READ; the file is in sequential access mode and a DELETE or REWRITE was attempted. There may have been an error or exception (such as an AT END) on the previous READ. The run continues, but nothing on the file will be changed.

User Action: Insert a READ statement before the DELETE or REWRITE statement, or change to ACCESS MODE IS DYNAMIC.

Further Information: See the discussion of the DELETE and REWRITE statements in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9240-- At line {text}, file {file} opened INPUT or I-O or EXTEND does not exist.

Condition Identifier: CB\$FILE_NOT_EXISTING Code: CB 9240

Description: The file was opened for input or input-output and it is not available to the run unit. It may not have been attached if it is a permanent file, or the file reference may be incorrect.

User Action: Check to see if the file has been attached and if the file reference has been specified correctly.

Further Information: See the SCL System Interface Usage manual for information on how to attach a file and how to specify a file reference correctly.

--FATAL CB 9241-- At line {text}, file {file} was empty and not previously opened.

Condition Identifier: CB\$NEW_FILE_EMPTY Code: CB 9241

Description: The copy was not performed because the input file contains no data.

User Action: Check that the correct input file is specified.

--WARNING CB 9245-- At line {text}, max_record_length for GIVING file {file} < that for SD.

Condition Identifier: CBE\$GIVING_RECORD_SHORT Code: CB 9245

Description: The maximum record length of the file, which is specified in the GIVING phrase of a SORT or MERGE statement, is less than that of the associated Sort Description entry. Any longer records are truncated.

User Action: Reconcile GIVING file and SD file record length specifications.

Further Information: See the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9250-- At line {text}, illegal reference modification. Leftmost-character-position is {text}. Length is {text}.

Condition Identifier: CBE\$ILLEGAL_REF_MOD Code: CB 9250

Description: Reference modification was invalid for one of the following reasons: either the leftmost-character-position or length was 0, negative, or larger than the item; or the sum of the leftmost-character-position and the length was larger than the size of the item + 1.

User Action: Correct program logic.

Further Information: See the discussion of reference modification in the COBOL Usage manual.

--WARNING CB 9255-- At line {text}, max_record_length for file {file} is larger than COBOL record area.

Condition Identifier: CBE\$MRL_TOO_BIG Code: CB 9255

Description: The MAXIMUM_RECORD_LENGTH attribute is larger than the record area for the file. It may have been set erroneously in a SET_FILE_ATTRIBUTE command. The record area size is used so some records may be truncated. Processing continues.

User Action: This message often occurs on a print file that is assigned to \$OUTPUT. Because \$OUTPUT has a MAXRL established by the operating system, any use of this file in a COBOL program will encounter the diagnostic when the program's record description is not equal to the system's value of 132 characters. You can set the MAXRL by a SET_FILE_ATTRIBUTE command.

--WARNING CB 9256-- At line {text}, max_record_length for file {file} is smaller than COBOL record area.

Condition Identifier: CBE\$MRL_TOO_SMALL Code: CB 9256

Description: The MAXIMUM_RECORD_LENGTH attribute is smaller than the record area for the file. It may have been set erroneously in a SET_FILE_ATTRIBUTE command. The records will be truncated. Processing continues.

User Action: Check the MAXIMUM_RECORD_LENGTH attribute by executing a DISPLAY_FILE_ATTRIBUTE command. Reconcile this with the record size specified in the program.

--FATAL CB 9260-- At line {text}, I-O error detected on file and no exception process provided.

Condition Identifier: CBE\$NO_IO_EXCEPTION Code: CB 9260

Description: An input-output error occurred on the file, and the program has no exception process to handle it.

User Action: Either add an AT END or INVALID KEY phrase to the input or output statement for the file, or specify an applicable USE AFTER EXCEPTION declarative, or specify a FILE STATUS data item for the file.

Further Information: See the discussion of the I-O status item in the COBOL Usage manual.

--FATAL CB 9265-- At line {text}, file {file} described as relative but is not.

Condition Identifier: CBE\$NOT_RELATIVE_FILE Code: CB 9265

Description: The file is described in the source code as ORGANIZATION IS RELATIVE, but the file processed is not a relative file.

User Action: Reconcile file description and file. Use DISPLAY_FILE_ATTRIBUTE command to verify the file attributes. It should be FO=BA and RT=F.

--FATAL CB 9270-- At line {text}, a READ or START on file {file}, which is not open for INPUT or I-O.

Condition Identifier: CBE\$READ_ON_OUTPUT Code: CB 9270

Description: A READ was attempted on a file open for output or extend. Either the file should be open for input or input-output or there is a logic error in the program.

User Action: Correct the OPEN statement or program logic.

Further Information: See the description of the OPEN statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9275-- At line {text}, READ NEXT on file {file}, which has no defined file position indicator.

Condition Identifier: CBE\$READ_WITH_UNDEFINED_FPI Code: CB 9275

Description: A READ NEXT attempted on a file that does not have a defined file position indicator. This can happen if the previous input-output operation on the file was unsuccessful.

User Action: A READ NEXT operation requires that a valid next record be established by the successful execution of an OPEN, READ, or START. Probably the previous READ or START operation was not successful. Correct program logic.

Further Information: See the discussion of READ NEXT in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9280-- At line {text}, RELEASE with record-name not in current SD.

Condition Identifier: CBE\$RELEASE_WRONG_RECORD Code: CB 9280

Description: A RELEASE record-name was executed and record-name is not within the Sort Description entry referenced in the active SORT statement. There is either a conflict in specifications or a program logic error.

User Action: Correct record-name in SD or RELEASE statement, or correct program logic.

Further Information: See the discussion of the RELEASE statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9285-- At line {text}, RELEASE with no sort active.

Condition Identifier: CBE\$RELEASE_NOT_IN_SORT Code: CB 9285

Description: A RELEASE statement was executed and no SORT statement is active. The RELEASE statement must be in a SORT INPUT PROCEDURE.

User Action: Correct program logic.

Further Information: See the discussion of the RELEASE statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9287-- At line {text}, RELEASE with no INPUT PROCEDURE active.

Condition Identifier: CBE\$RELEASE_NO_INPUT_PROC Code: CB 9287

Description: A RELEASE statement was executed and no INPUT PROCEDURE is active. The RELEASE statement must be within the range of a SORT INPUT PROCEDURE.

User Action: Move the RELEASE statement to within the SORT INPUT PROCEDURE.

Further Information: See the discussion of the RELEASE statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9290-- At line {text}, RETURN after AT END.

Condition Identifier: CBE\$RETURN_AFTER_AT_END Code: CB 9290

Description: A RETURN statement executed after the previous RETURN produced an at-end condition. Possible program logic error.

User Action: Correct program logic.

Further Information: See the discussion of the RETURN statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9295-- At line {text}, RETURN with file-name not current SD.

Condition Identifier: CBE\$RETURN_WRONG_FILE Code: CB 9295

Description: A RETURN file-name executed and file-name is not in the Sort Description entry referenced in the active SORT or MERGE statement. There is either a conflict in specifications or a program logic error.

User Action: Correct file-name in SD or RETURN statement, or correct program logic.

Further Information: See the discussion of the RETURN statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9300-- At line {text}, RETURN with no sort or merge active.

Condition Identifier: CBE\$RETURN_NOT_IN_SM Code: CB 9300

Description: A RETURN statement was executed and no SORT or MERGE statement is active. The RETURN must be in the range of a SORT or MERGE OUTPUT PROCEDURE.

User Action: Correct program logic.

Further Information: See the discussion of the RETURN statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9303-- At line {text}, RETURN with no OUTPUT PROCEDURE active.

Condition Identifier: CBE\$RETURN_NO_OUTPUT_PROC Code: CB 9303

Description: A RETURN statement was executed and no OUTPUT PROCEDURE was active. A RETURN statement must be within the range of an active OUTPUT PROCEDURE that was invoked by a SORT or MERGE statement.

User Action: Move the RETURN statement to within a SORT or MERGE OUTPUT PROCEDURE.

Further Information: See the discussion of the RETURN statement in the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9305-- At line {text}, {text} executed with a {text} already active.

Condition Identifier: CBE\$SM_WITHIN_SM Code: CB 9305

Description: A SORT or MERGE statement executed with a SORT or MERGE already active.

User Action: Remove the SORT or MERGE statement from the INPUT PROCEDURE or OUTPUT PROCEDURE.

Further Information: See the Sort/Merge chapter in the COBOL Usage manual.

--FATAL CB 9310-- The sentence beginning at line {text} contained a fatal compilation error.

Condition Identifier: CBE\$SOURCE_STATEMENT_ERROR Code: CB 9310

Description: A syntax error during compilation was detected on the indicated line.

User Action: Correct the error and recompile the program.

--WARNING CB 9315-- At line {text}, STOP RUN not encountered in program.

Condition Identifier: CBE\$NO_STOP_RUN Code: CB 9315

Description: A program fell through the last statement and no STOP RUN had been executed. The run is terminated as if a STOP RUN were executed.

User Action: Add STOP RUN as the last executable statement.

Further Information: See the discussion of the STOP RUN statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9320-- At line {text}, Subscript {text} of value {text} is out of range.

Condition Identifier: CBE\$SUBSCRIPT_ERROR Code: CB 9320

Description: The subscript whose level is given is 0, negative, or larger than the maximum value allowed for the associated OCCURS clause.

User Action: Correct subscript or program logic.

Further Information: See the discussion of the OCCURS clause in the COBOL Usage manual.

--FATAL CB 9325-- At line {text}, the report footing and the last page footing overlap for report {text}.

Condition Identifier: CBE\$REPORT_FOOTING_OVERLAP Code: CB 9325

Description: The report footing and the last page footing presented overlap due to the page specifications.
The report footing is suppressed.

User Action: Correct the page specifications.

Further Information: See the Report Writer chapter in the COBOL Usage manual.

--FATAL CB 9330-- At line {text}, the report heading and the first page heading overlap for report {text}.

Condition Identifier: CBE\$REPORT_HEADING_OVERLAP Code: CB 9330

Description: The report heading and the first page heading presented overlap due to the page specifications.
The report heading is suppressed.

User Action: Correct the page specifications.

Further Information: See the Report Writer chapter in the COBOL Usage manual.

--WARNING CB 9335-- At line {text}, report {text} already initiated.

Condition Identifier: CBE\$REPORT_ALREADY_INITIATED Code: CB 9335

Description: The report is already initiated. The latest INITIATE statement is ignored.

User Action: Remove duplicate INITIATE statement or otherwise correct program.

Further Information: See the discussion of the INITIATE statement in the Report Writer chapter of the COBOL Usage manual.

--WARNING CB 9340-- At line {text}, report {text} not initiated when TERMINATE executed.

Condition Identifier: CBE\$REPORT_NOT_INITIATED Code: CB 9340

Description: The report has not been initiated by an INITIATE statement. The current TERMINATE statement is ignored.

User Action: Add INITIATE statement or otherwise correct program.

Further Information: See the discussion of the INITIATE statement in the Report Writer chapter of the COBOL Usage manual.

--WARNING CB 9341-- At line {text}, report {text} not initiated when GENERATE executed.

Condition Identifier: CBE\$REPORT_NOT_INITIATED_GEN Code: CB 9341

Description: The report has not been initiated by an INITIATE statement. The report will be initiated prior to execution of this GENERATE statement.

User Action: Add INITIATE statement or otherwise correct program.

Further Information: See the discussion of the INITIATE statement in the Report Writer chapter of the COBOL Usage manual.

--FATAL CB 9342-- At line {text}, error encountered in report spacing routine for report {text}.

Condition Identifier: CBE\$REPORT_SPACING_ERROR Code: CB 9342

Description: An error was discovered in the spacing routine for this report. This is a system error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL CB 9345-- At line {text}, unaltered GO TO executed.

Condition Identifier: CBE\$UNALTERED_GO_TO Code: CB 9345

Description: A "GO TO." statement that has not been set with an ALTER statement was executed.

User Action: Delete the statement or set it with an ALTER statement.

Further Information: See the discussion of the GO TO and ALTER statements in the Procedure Division chapter of the COBOL Usage manual.

--WARNING CB 9350-- At line {text}, max_record_length for USING FILE {file} > that for SD.

Condition Identifier: CBE\$USING_MRL_TOO_BIG Code: CB 9350

Description: The maximum record length of the file, which is specified in the USING phrase of a SORT statement, is larger than the maximum record length of the associated Sort Description entry. Any records that are larger than the SD length are truncated with no indication given.

User Action: Reconcile the USING file and the SD file record length specifications.

Further Information: See the Sort/Merge chapter of the COBOL Usage manual.

--FATAL CB 9355-- At line {text}, a {text} attempted on file {file}, which is not open.

Condition Identifier: CBE\$IO_ON_UNOPENED_FILE Code: CB 9355

Description: An input-output operation was attempted on a file that is not open. The OPEN request for the file may have failed for some reason or there is a logic error in the program. If run continuation is selected (as with CBP\$IO_ERROR_NO_ABORT), the operation will not be successful.

User Action: Correct program logic.

Further Information: Check the error that occurred on the OPEN statement. See the discussion of the OPEN statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9360-- At line {text}, the open mode of file {file} does not support the attempted {text}.

Condition Identifier: CBE\$OUTPUT_OP_NOT_OPEN_OUTPUT Code: CB 9360

Description: An output operation was attempted on a file that is not open in a mode that allows the operation. For a sequential file, it must be OPEN EXTEND or OUTPUT. For any other file organization, it must be OPEN EXTEND, I-O, or OUTPUT.

User Action: Change the OPEN statement to specify the correct open mode.

Further Information: See the description of the OPEN statement in the Procedure Division chapter of the COBOL Usage manual.

--WARNING CB 9365-- At line {text}, I-O error on paragraph trace file {file}. I-O error status follows.

Condition Identifier: CBE\$PTRACE_IO_ERROR Code: CB 9365

Description: An input-output error occurred on the paragraph trace file. Paragraph trace is turned off for the remainder of execution. Processing continues.

User Action: Examine the input-output error status to determine corrective action.

--WARNING CB 9370-- At line {text}, program services request error while processing paragraph trace. Program services error status follows.

Condition Identifier: CBE\$PTRACE_PMP_ERROR Code: CB 9370

Description: A request to obtain the elapsed CP time from program services has failed. Paragraph trace is turned off for the remainder of execution.

User Action: Examine the program services error status to determine corrective action.

--FATAL CB 9375-- At line {text}, a {text} operation was attempted on file {file}, but the file is not open for I-O.

Condition Identifier: CBE\$IO_MODE_ERROR Code: CB 9375

Description: The indicated operation requires that the file be opened in the input-output mode, and the file was not opened in that mode. There is probably a logic error in the program.

User Action: Correct program logic and make sure that the proper form of OPEN is executed before the indicated operation.

Further Information: See the discussion of the OPEN statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9380-- At line {text}, a READ NEXT operation was attempted on file {file}, but no valid next record has been established.

Condition Identifier: CBE\$READ_NEXT_NOT_LEGAL Code: CB 9380

Description: A READ NEXT operation requires that a valid next record be established by the successful execution of an OPEN, READ, or START. Probably the previous READ or START operation was not successful.

User Action: Correct program logic.

Further Information: See the discussion of READ NEXT in the COBOL Usage manual.

--FATAL CB 9385-- At line {text}, file {file}: An unexpected access method error returned. It follows.

Condition Identifier: CBE\$CRM_ERROR Code: CB 9385

Description: The access method returned an unexpected error status. This could be a system error or a user error.

User Action: Take appropriate action as indicated by the nature of the error.

--FATAL CB 9390-- At line {text}, system request {text} resulted in an error status which follows.

Condition Identifier: CBE\$SYSTEM_ERROR_STATUS Code: CB 9390

Description: The indicated system request resulted in an unexpected error status. This indicates a system error. Normally, the operation will continue as if no error had occurred.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL CB 9395-- At line {text}, file {file} is being opened for {text} but ACCEPT or DISPLAY already opened the same file with a different mode.

Condition Identifier: CBE\$OPEN_MODE_VS_ACC_DIS Code: CB 9395

Description: The file is being opened as indicated, but the first reference to the file was by an ACCEPT or DISPLAY statement, and that reference caused the file to be opened in a different open mode.

User Action: Either change the implementor-name for the file or remove the ACCEPT or DISPLAY statement.

Further Information: See the discussion of the ACCEPT and DISPLAY statements in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9397-- At line {text}, DISPLAY UPON CONSOLE failed due to operator.

Condition Identifier: CBE\$OPERATOR_ERROR Code: CB 9397

Description: Either the operator was not at the console, or the operator failed to clear the previous message. The message could not be sent.

User Action: Contact operator.

--FATAL CB 9398-- Error with {text} during sorting or merging relative files.

Condition Identifier: CBE\$RELATIVE_SORT Code: CB 9398

Description: An error occurred while processing a relative file to get it ready to sort.

User Action: Examine the file.

--CATASTROPHIC CB 9400-- At line {text}, the COBOL system error "{text}" occurred in the module {text}. Notify your CDC analyst.

Condition Identifier: CBE\$RUN_TIME_SYSTEM_ERROR Code: CB 9400

Description: The indicated system error was detected by the COBOL run-time library. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--WARNING CB 9415-- At line {text}, file {file} not closed at STOP RUN - closed by COBOL.

Condition Identifier: CB\$FILE_OPEN_AT_STOP_RUN Code: CB 9415

Description: The indicated file was still open when the STOP RUN statement was executed. The file has been closed by COBOL.

User Action: Correct program logic.

Further Information: See the discussion of the STOP RUN statement in the Procedure Division chapter of the COBOL Usage manual.

--FATAL CB 9416-- Parameters specified on execution command for COBOL program are incorrect.

Condition Identifier: CB\$RUNTIME_CRACKER_ERROR Code: CB 9416

Description: The parameters specified on the execution command for a COBOL program are incorrect.

User Action: Inspect the syntax of the parameter specification and ensure that the file names being equivalenced exist as implementor names within the COBOL program being executed.

Further Information: See the discussion of the execution parameters in the COBOL Usage manual.

--FATAL CB 9420-- At line {text}, ACCEPT/DISPLAY attempted on file, {file}, whose file organization is not sequential.

Condition Identifier: CB\$ACCEPT_DISPLAY_INVALID_ORG Code: CB 9420

Description: can use ACCEPT and DISPLAY statements only on sequential files.

User Action: Check to see if you have indicated the correct file name in the ACCEPT or DISPLAY statement.

Further Information: See the discussion of the ACCEPT and DISPLAY statements in the COBOL Usage manual.

--FATAL CB 9425-- At line {text}, file {file} was opened, and it did not contain the alternate key {text} defined in the COBOL program.

Condition Identifier: CB\$INVALID_ALTERNATE_KEY Code: CB 9425

Description: The indicated alternate key is not defined for the file.

User Action: Inspect the file description by using the DISPLAY_FILE_ATTRIBUTES command and inspect the COBOL program.

Further Information: See the discussion of the ALTERNATE RECORD KEY Clause in the COBOL Usage manual.

--FATAL CB 9430-- At line {text}, the parameter used to specify the lock intent must be "EXCLUSIVE_ACCESS", "PRESERVE_CONTENT", or "PRESERVE_ACCESS_AND_CONTENT".

Condition Identifier: CB\$INVALID_LOCK_INTENT_OPTION Code: CB 9430

Description: The USING phrase in the CBP\$SET_KEY_LOCKING_OPTIONS, CBP\$LOCK_KEY, and CBP\$LOCK_FILE calls contains a parameter specifying the kind of lock to request. That parameter must be "EXCLUSIVE_ACCESS", "PRESERVE_CONTENT", or "PRESERVE_ACCESS_AND_CONTENT".

User Action: Change the parameter to be "EXCLUSIVE_ACCESS", "PRESERVE_CONTENT", or "PRESERVE_ACCESS_AND_CONTENT".

--FATAL CB 9435-- At line {text}, the parameter used to specify whether records are to be locked must be "LOCK" or "NO_LOCK".

Condition Identifier: CB\$INVALID_LOCK_SEL_OPTION Code: CB 9435

Description: The second parameter listed in the USING phrase of the CBP\$SET_KEY_LOCKING_OPTIONS call must be either "LOCK" or "NO_LOCK".

User Action: Change the second parameter in the USING phrase of the CBP\$SET_KEY_LOCKING_OPTIONS call to be either "LOCK" or "NO_LOCK".

--FATAL CB 9440-- At line {text}, the parameter used to specify unlock control must be "AUTOMATIC_UNLOCK" or "WAIT_FOR_UNLOCK".

Condition Identifier: CBE\$INVALID_UNLOCK_CTRL_OPTION Code: CB 9440

Description: The USING phrase in the CBP\$SET_KEY_LOCKING_OPTIONS and CBP\$LOCK_KEY calls contains a parameter specifying how key values are to be unlocked. This parameter must be either "AUTOMATIC_UNLOCK" or "WAIT_FOR_UNLOCK".

User Action: Change the parameter to be either "AUTOMATIC_UNLOCK" or "WAIT_FOR_UNLOCK".

--FATAL CB 9445-- At line {text}, the parameter used to specify the keys to be unlocked must be "ALL" or "ONE".

Condition Identifier: CBE\$INVALID_UNLOCK_SEL_OPTION Code: CB 9445

Description: The second parameter listed in the USING phrase of the CBP\$UNLOCK_KEY call must be either "ALL" or "NONE".

User Action: Change the second parameter in the USING phrase of the CBP\$UNLOCK_KEY call to be either "ALL" or "NONE".

--FATAL CB 9450-- At line {text}, the parameter used to specify whether to wait for a lock to become available must be "WAIT_FOR_LOCK" or "NO_WAIT_FOR_LOCK".

Condition Identifier: CBE\$INVALID_WAIT_OPTION Code: CB 9450

Description: The USING phrase of the CBP\$SET_KEY_LOCKING_OPTIONS, CBP\$LOCK_KEY, and CBP\$LOCK_FILE calls specifies whether to wait for a record to become available. That parameter must be either "WAIT_FOR_LOCK" or "NO_WAIT_FOR_LOCK".

User Action: Change the parameter to be either "WAIT_FOR_LOCK" or "NO_WAIT_FOR_LOCK".

--FATAL CB 9455-- At line {text}, file {file} is required to be a mass storage file.

Condition Identifier: CBE\$MASS_STORAGE_FILE_REQUIRED Code: CB 9455

Description: COBOL program assigned a file with SEGMENT-ACCESS organization to a NOS/VE file. At run time, that NOS/VE file is not a mass storage file.

User Action: Be sure that the file with SEGMENT-ACCESS organization is assigned to a mass storage file.

--FATAL CB 9457-- The size of the table passed to CBP\$DEFINE_APPLIC_UNIT_TABLE must be from 1 to 63 full words.

Condition Identifier: CBE\$APPLIC_ARRAY_ERROR Code: CB 9457

Description: The USING phrase of CBP\$DEFINE_APPLIC_UNIT_TABLE has defined an incorrect application table size. The application table size must be from 1 to 63 full words.

User Action: Correct the table size so that it is from 1 to 63 given as an integer literal or an identifier defined as a full word COMP item.

Further Information: See the discussion of Unit Measured Application Accounting in the Inter-Program Communication chapter of the COBOL Usage manual.

--FATAL CB 9459-- The first parameter for a call to CBP\$PREFERRED_OPEN_SHARE_MODES must be a quoted literal containing the applicable of the following choices specified in uppercase letters and separated by spaces: APPEND, READ, MODIFY, SHORTEN, and EXECUTE.

Condition Identifier: CBE\$OPEN_SHARE_MODES_INCORRECT Code: CB 9459

Description: The parameter is not valid.

User Action: Change the parameter to one of the valid choices.

--FATAL CB 9460-- At line {text}, TERMINAL cannot be accessed; the current job must be of interactive mode.

Condition Identifier: CBE\$REQUEST_TERMINAL_FAILED Code: CB 9460

Description: The user cannot use TERMINAL or TERMINAL-C as the implementer name.

User Action: Change to \$IMPUT or a data file.

--FATAL CB 9461-- At line {text}, ACCEPT FROM CONSOLE failed due to operator.

Condition Identifier: CBE\$ACCEPT_FROM_CONSOLE_FAILED Code: CB 9461

Description: Failure due to operator error.

User Action: Correct and rerun program.

--FATAL CB 9462-- At line {text}, file {file} cannot be opened for ACCEPT statement.

Condition Identifier: CBE\$ACCEPT_FILE_OPEN_FAILED Code: CB 9462

Description: The user tries to do ACCEPT...FROM on a file that is already open.

User Action: Correct the program.

--FATAL CB 9999-- COBOL internal error: run time diagnostic number too high.

Condition Identifier: CBE\$LAST_ERROR Code: CB 9999

Description: During the execution of COBOL object code or COBOL generated code, a COBOL runtime routine is trying to issue a diagnostic with a diagnostic number that is too high.

User Action: See the site analyst.

--FATAL CF 0-- Input and Output files are the same.

Condition Identifier: CFE\$INPUT_OUTPUT_SAME Code: CF 0

Description: The INPUT and OUTPUT parameters must specify different files. Formatting does not occur if the same file name is specified for input and output.

User Action: Change the OUTPUT file name specified by the OUTPUT parameter and execute FORMAT_CYBIL_SOURCE again.

--WARNING CF 1-- No matching {text} for {text} on line {text}.

Condition Identifier: CFE\$UNBALANCED_BLOCK_STRUCTURE Code: CF 1

Description: The block structure is not satisfied (e.g., no matching WHILE for WHILEND). The line number indicates a location within the input file.

User Action: Correct the input file and format again.

--WARNING CF 2-- Unmatched parenthesis on line {text}: {text}

Condition Identifier: CFE\$UNMATCHED_PARENS Code: CF 2

Description: The source line in error is written to the error listing. The number of right parentheses does not match the number of left parentheses. The line number indicates a location in the source file.

User Action: Correct the input file and format again.

--WARNING CF 3-- Unmatched brackets on line {text}: {text}

Condition Identifier: CFE\$UNMATCHED_BRACKETS Code: CF 3

Description: The source line in error is written to the error listing. The number of right brackets does not match the number of left brackets.

User Action: Correct the input file and format again.

--WARNING CF 5-- Expecting {text} found {text} on line {text}: {text}

Condition Identifier: CFE\$EXPECTING_FOUND_ERROR Code: CF 5

Description: The source line in error is written to the error listing. A CYBIL syntax error has been detected. The line number indicates a location within the source file.

User Action: The source has been fully formatted. You may want to correct the CYBIL syntax error in the output file. You might also correct the input file and reformat.

--WARNING CF 6-- Expecting {text} <= integer value <= {text} , found {text} on line {text}: {text}

Condition Identifier: CF\$INTEGER_RANGE_ERROR Code: CF 6

Description: The source line in error is written to the error listing. This error occurs when an FMT toggle parameter does not fall within the acceptable range of values. The line number indicates a location within the source file.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a description of the FMT directive and all toggles defined for it.

--INFORMATIVE CF 7-- Line {text} is too long for current margin values. {text}

Condition Identifier: CF\$LINE_TOO_LONG Code: CF 7

Description: The source line in error is written to the error listing. Formatting has been turned off, and the source line exceeds the right margin.

User Action: Informative message; you may choose to ignore it.

--FATAL CF 8-- Undefined format option selected {text}.

Condition Identifier: CF\$ILLEGAL_FORMAT_OPTION Code: CF 8

Description: You have specified a format option that is not defined.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a description of the FORMAT_CYBIL_SOURCE command and formatting options.

--WARNING CF 9-- Length of tab character {text} must be <= {text} on line {text}: {text}

Condition Identifier: CF\$TAB_CHARACTER_ERROR Code: CF 9

Description: The source line in error is written to the error listing. The length of the tab character identifier exceeds the maximum string length. The tab string can consist of 1 to 8 characters.

User Action: Correct the FMT directive in error and format again.

--WARNING CF 11-- Unbalanced block structure -- possible {text} missing for {text} on line {text} or {text} missing for {text} on line {text}.

Condition Identifier: CF\$UNBALANCED_RANGE_MESS Code: CF 11

Description: The begin and end statements for a block structure do not match. The formatter detected a mismatch between the two lines indicated. In some cases the formatter can not determine which of two structures is unbalanced. This message may be issued against a line that is not in error.

User Action: Examine the input file to determine if the line indicated is in error. Correct the line in error and format again.

--WARNING CF 12-- Syntax error -- identifier too long or unrecognized character on line {text}: {text}

Condition Identifier: CF\$SYNTAX_ERROR Code: CF 12

Description: A CYBIL identifier is greater than 31 characters, or an illegal symbol was encountered in the CYBIL text.

User Action: Correct the input file and format again.

--INFORMATIVE CF 13-- Token {text} on line {text} is too long for current margin values. {text}

Condition Identifier: CF\$TOKEN_TOO_LONG Code: CF 13

Description: The indicated token is too long for the current width of the line (for example, a 15-character token attempting to fit in a 12-character line). The line number refers to the input file. The last text parameter is the line containing the token.

User Action: Correct the margin values on the input file and format again.

--FATAL CF 100-- Insufficient central memory available.

Condition Identifier: CFE\$NO_CENTRAL_MEMORY Code: CF 100

Description: When the message was issued, memory limits had been exceeded.

User Action: Check to see if the formatter had a problem. If not, release memory and try again.

--INFORMATIVE CF 200-- Token beginning with {text} on line {text} is > {text} characters. Token truncated to {text} characters.

Condition Identifier: CFE\$TOKEN_LENGTH_ERROR Code: CF 200

Description: A literal token exceeded 255 characters.

User Action: If truncation is unacceptable, change the code and format again.

--ERROR CL 1-- Version {text} for an SCL table or specification declaration is not supported.

Condition Identifier: CLE\$BAD_DECLARATION_VERSION Code: CL 1

Description: The version of a parameter description table (PDT) or type specification is not one that is known or supported.

User Action: Use the GENERATE_PDT command to produce a correct version of the PDT or type specification.

--ERROR CL 2-- Bad clt\$type_specification for KEYWORD type.

Condition Identifier: CLE\$BAD_KEYWORD_TYPE_SPEC Code: CL 2

Description: Self-explanatory.

User Action: Use the GENERATE_PDT command to produce a correct version of the PDT or type specification.

--ERROR CL 3-- The parameter description table passed to clp\$evaluate_sub_parameters may not contain VAR (pass by reference) parameters.

Condition Identifier: CLE\$VAR_SUB_PARAMS_NOT_ALLOWED Code: CL 3

Description: Self-explanatory.

User Action: Use the GENERATE_PDT command to produce a correct version of the PDT or type specification.

--ERROR CL 4-- A clt\$type_description is improperly constructed.

Condition Identifier: CLE\$BAD_TYPE_DESCRIPTION Code: CL 4

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 5-- ALL must be used alone for parameter {text}.

Condition Identifier: CLE\$ALL_MUST_BE_USED_ALONE Code: CL 5

Description: The keyword value, ALL, must be used alone. It cannot be used within a range or list specification.

User Action: Either remove the ALL specification or specify only the ALL keyword value.

--ERROR CL 6-- Improper value supplied for REPRESENTATION_OPTION parameter of CLP\$CONVERT_DATA_TO_STRING request.

Condition Identifier: CLE\$BAD_DATA_REP_OPTION Code: CL 6

Description: An incorrect value was entered for the representation option parameter.

User Action: Correct the representation option parameter and retry.

--ERROR CL 7-- A clt\$data_value is improperly constructed.

Condition Identifier: CLE\$BAD_DATA_VALUE Code: CL 7

Description: All of the necessary fields of a CLT\$DATA_VALUE record have not been correctly "filled in", probably by a function processor.

User Action: Make sure that all relevant fields of the record are set to an appropriate value.

--WARNING CL 8-- The following tasks have already completed: {text}

Condition Identifier: CLE\$TASK_ALREADY_COMPLETE Code: CL 8

Description: Self-explanatory.

User Action: No action required.

--ERROR CL 9-- The following tasks are not known to the requesting task: {text}

Condition Identifier: CLE\$TASK_NOT_FOUND Code: CL 9

Description: The task name specified does not exist or was not originated by the task making the request.

User Action: Check the spelling or return to the task that originated the task the user is inquiring about.

--INFORMATIVE CL 10-- Waiting for asynchronous task(s) to terminate.

Condition Identifier: CLE\$AWAITING_TASK_TERMINATION Code: CL 10

Description: The current task is waiting for an asynchronous task to terminate.

User Action: None.

--INFORMATIVE CL 11-- Child task for application {text} could not be found in the application task link.

Condition Identifier: CLE\$BAD_APPLICATION_TASK_LINK Code: CL 11

Description: This is an internal error.

User Action: Notify a site analyst to submit a PSR.

--FATAL CL 12-- --FATAL CL 12-- decremented because it was already zero. (PVA of block = {text}).

Condition Identifier: CLE\$BLOCK_ACCESS_COUNT_ERROR Code: CL 12

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 13-- Cannot access application counter array for application {text}.

Condition Identifier: CLE\$CANNOT_ACCESS_UNIT_ARRAY Code: CL 13

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 14-- A {text} value cannot be represented as a clt\$value.

Condition Identifier: CLE\$UNEXPECTED_VALUE_TYPE Code: CL 14

Description: Self-explanatory.

User Action: Convert the command or function to use clp\$evaluate_parameters, etc.

--ERROR CL 15-- File {file} is empty.

Condition Identifier: CLE\$EMPTY_FILE Code: CL 15

Description: The specified file that was expected to contain data is empty.

User Action: Ensure that the correct file is specified.

--ERROR CL 16-- A clt\$value is improperly constructed.

Condition Identifier: CLE\$BAD_CLT\$VALUE Code: CL 16

Description: All of the necessary fields of a CLT\$VALUE record have not been correctly "filled in", probably by a function processor.

User Action: Make sure that all relevant fields of the record are set to an appropriate value.

--ERROR CL 17-- A clt\$parameter_list is improperly constructed.

Condition Identifier: CLE\$BAD_PARAMETER_LIST Code: CL 17

Description: A command or function's parameter list passed to clp\$evaluate_parameters (or clp\$scan_parameter_list on older systems) is incorrect.

User Action: The first item in the CLT\$PARAMETER_LIST sequence must be a CLT\$STRING_SIZE that specifies the number of characters in the text of the parameter list. This is followed by the parameter list's text itself. Note that when the SCL interpreter calls a command or function processor calls a command or function processor, the contents of the CLT\$PARAMETER_LIST sequence differs from this structure. Instead the contents is a "code" that permits faster analysis of the parameter list. This "code" can only be correctly interpreted by clp\$evaluate_parameters (or clp\$scan_parameter_list) if the calling command or function processor was called directly by the SCL interpreter. Clp\$get_parameter_list_text can handle a CLT\$PARAMETER_LIST in either format.

--ERROR CL 18-- A clt\$variable_value is improperly constructed.

Condition Identifier: CLE\$BAD_CLT\$VARIABLE_VALUE Code: CL 18

Description: All of the necessary fields of a CLT\$VARIABLE_VALUE record have not been correctly "filled in", probably by a function processor.

User Action: Make sure that all relevant fields of the record are set to an appropriate value.

--ERROR CL 19-- Conflicting options were specified for function {text}.

Condition Identifier: CLE\$CONFLICTING_OPTIONS_SPEC Code: CL 19

Description: Both "NAMES" and "PATHS" cannot be specified for the OPTIONS parameter of the function.

User Action: Supply either the "NAMES" or "PATHS" parameter.

--WARNING CL 20-- {text} encountered EOI.

Condition Identifier: CLE\$ENCOUNTERED_EOI Code: CL 20

Description: The end of information of the file was encountered before it was expected.

User Action: Ensure that the file contains all required data.

--ERROR CL 21-- A clt\$parameter_description_table generated for command can't be used by the processor for a function; and one generated for a function can't be used by the processor for a command.

Condition Identifier: CLE\$PDT_PROCESSOR_MISMATCH Code: CL 21

Description: Self-explanatory.

User Action: Correct the source of the pdt and regenerate it using the GENERATE_PDT command.

--ERROR CL 25-- A clt\$type_specification is improperly constructed.

Condition Identifier: CLE\$BAD_TYPE_SPECIFICATION Code: CL 25

Description: Self-explanatory.

User Action: Use the GENERATE_PDT command to produce a correct version of the type specification.

--ERROR CL 26-- End of input encountered while processing {text} declaration.

Condition Identifier: CLE\$EOI_IN_DECLARATION Code: CL 26

Description: Self-explanatory.

User Action: Complete the parameter description table (PDT) or correct the type declaration and retry.

--ERROR CL 27-- {text} was issued for file {file} which has never been opened.

Condition Identifier: CLE\$FILE_NEVER_OPENED Code: CL 27

Description: A file must be written before it can be read.

User Action: None.

--ERROR CL 28-- Unable to call procedure to read input for {text} declaration.

Condition Identifier: CLE\$UNABLE_TO_CALL_INPUT_PROC Code: CL 28

Description: The procedure specified to get the lines for a parameter description table (PDT) or type declaration could not be called. Probably it could not be loaded.

User Action: Check the residence of the specified procedure.

--ERROR CL 29-- {text} is not supported.

Condition Identifier: CLE\$NOT_SUPPORTED Code: CL 29

Description: The facility named in the message is not supported by the system.

User Action: None.

--ERROR CL 30-- {text} refer to the same file.

Condition Identifier: CLE\$FILE_REFERENCE_CONFLICT Code: CL 30

Description: A file was specified to be used for two or more purposes on a command.

User Action: Use a different file for each purpose.

--ERROR CL 35-- "{text}" may not be used for a substitution mark.

Condition Identifier: CLE\$IMPROPER_SUBSTITUTION_MARK Code: CL 35

Description: Only printable characters that may not be used in SCL names are acceptable for use as a substitution mark.

User Action: Use one of the allowable characters, for example "&".

--ERROR CL 37-- A substitution mark may only be use in a command for an asynchronous task.

Condition Identifier: CLE\$IMPROPER_USE_OF_SUBST_MARK Code: CL 37

Description: Use of a substitution mark with a TASK/TASKEND command is only supported for asynchronous tasks , i.e. if the TASK_NAME parameter is used.

User Action: For a synchronous task, direct variable and parameter references are used.

--ERROR CL 40-- Parameters given are incompatible with value of parameter {text}.

Condition Identifier: CLE\$INCOMPATIBLE_PARAMS_GIVEN Code: CL 40

Description: The given parameters are incompatible with the value of the specified parameters.

User Action: Consult the command description for information about which parameters can be used together.

--ERROR CL 42-- Application {text} attempted to define more than one application counter array.

Condition Identifier: CLE\$MULTIPLE_APPLIC_UNIT_ARRAYS Code: CL 42

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 43-- Application {text} terminated with a negative value in counter {text} of the application counter array.

Condition Identifier: CLE\$NEGATIVE_APPLICATION_UNITS Code: CL 43

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 45-- File {file} is not list or legible.

Condition Identifier: CLE\$NOT_LIST_LEGIBLE Code: CL 45

Description: The requested operation requires a file whose FILE_CONTENTS attribute is LIST or LEGIBLE (unknown is treated a LEGIBLE).

User Action: Use a file of the appropriate type.

--ERROR CL 50-- NONE must be used alone for parameter {text}.

Condition Identifier: CLE\$NONE_MUST_BE_USED_ALONE Code: CL 50

Description: The keyword value, NONE, must be used alone. It cannot be used within a range or list specification.

User Action: Either remove the NONE specification or specify only the NONE keyword value.

--ERROR CL 55-- {text} not yet implemented.

Condition Identifier: CLE\$NOT_YET_IMPLEMENTED Code: CL 55

Description: The facility named in the message is not available in the present version of the system. It will become available in a later version.

User Action: None.

--ERROR CL 60-- Redundant values given for parameter {text}.

Condition Identifier: CLE\$REDUNDANCY_IN_SELECTIONS Code: CL 60

Description: The same value was given more than once in a list of values for a parameter.

User Action: Select the duplicated value only once.

--INFORMATIVE CL 62-- Asynchronous application tasks were terminated by SCL when application {text} ended.

Condition Identifier: CLE\$TERMINATED_APPLICATION_TASK Code: CL 62

Description: A subtask of an application has been terminated.

User Action: None.

--ERROR CL 65-- Caller ring # too high to set minimum working set.

Condition Identifier: CLE\$UNABLE_TO_SET_MINWS Code: CL 65

Description: You do not have sufficient privilege to change the minimum working set attribute of the job.

User Action: No action is required.

--ERROR CL 67-- Caller ring # too high to set page aging interval.

Condition Identifier: CLE\$UNABLE_TO_SET_PAI Code: CL 67

Description: You do not have sufficient privilege to change the page aging interval attribute of the job.

User Action: No action is required.

--ERROR CL 68-- Caller ring # too high to set cyclic aging interval.

Condition Identifier: CLE\$UNABLE_TO_SET_CAI Code: CL 68

Description: You do not have sufficient privilege to change the cycle aging interval attribute of the job.

User Action: No action is required.

--ERROR CL 69-- User is already logged in.

Condition Identifier: CLE\$USER_ALREADY_LOGGED_IN Code: CL 69

Description: The LOGIN command can only be used as the first command of a batch job.

User Action: Start a new job.

--FATAL CL 70-- Table overflow: {text}.

Condition Identifier: CLE\$TABLE_OVERFLOW Code: CL 70

Description: An internal system table does not contain enough space in it to complete the requested operation.

User Action: Reduce the request.

--FATAL CL 85-- --FATAL CL 85-- outstanding references to it. (PVA of unfreed block = {text}.)

Condition Identifier: CLE\$UNABLE_TO_FREE_BLOCK Code: CL 85

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--FATAL CL 90-- Unexpected call to {text}.

Condition Identifier: CLE\$UNEXPECTED_CALL_TO Code: CL 90

Description: A command or request was used in the wrong context. Another command or request must be used prior to the desired one.

User Action: Consult the description of the command or request.

--ERROR CL 95-- Same number of values must be supplied for parameters {text}.

Condition Identifier: CLE\$VALUE_COUNTS_UNEQUAL Code: CL 95

Description: The same number of values were not supplied for the specific parameters.

User Action: Supply the same number of values for invalid parameters.

--ERROR CL 96-- The system_prolog command can only be executed from within the system prolog.

Condition Identifier: CLE\$SYSTEM_PROLOG_NOT_ALLOWED Code: CL 96

Description: A system prolog is not allowed.

User Action: No user action required.

--ERROR CL 99-- Work area overflow in {text}.

Condition Identifier: CLE\$WORK_AREA_OVERFLOW Code: CL 99

Description: The work area passed to the designated routine was not large enough to complete the requested operation.

User Action: Increase the size of the work area.

--ERROR CL 105-- Alphabetic character in exponent part of real number: {text}.

Condition Identifier: CLE\$ALPHA_CHAR_IN_EXPONENT Code: CL 105

Description: An exponent can contain only digits.

User Action: Remove the letter from the exponent.

--ERROR CL 110-- Alphabetic character in fraction part of real number: {text}.

Condition Identifier: CLE\$ALPHA_CHAR_IN_FRACTION Code: CL 110

Description: The fractional part of a real token can contain only digits.

User Action: Remove the letter from the fractional part.

--ERROR CL 115-- Alphabetic character in number: {text}.

Condition Identifier: CLE\$ALPHA_CHAR_IN_NUMBER Code: CL 115

Description: A number cannot contain letters. If the number is in hexadecimal, the A, B, C, D, E, and F are interpreted as numbers; however, the number must begin with a digit and end with the radix specification (16) [for example, 0FF(16)].

User Action: Either remove the letters from the number or add a hexadecimal radix specifier.

--ERROR CL 120-- Digit too large for radix of integer: {text}.

Condition Identifier: CLE\$DIGIT_TOO_LARGE Code: CL 120

Description: A digit within the number is greater than or equal to the radix value.

User Action: Correct the number or radix specification.

--FATAL CL 125-- Expecting digit in number: {text}.

Condition Identifier: CLE\$EXPECTING_DIGIT Code: CL 125

Description: A number (integer or real) can contain only digits.

User Action: Remove the character from the number that is not a digit.

--ERROR CL 130-- Real number exponent too large: {text}.

Condition Identifier: CLE\$EXPONENT_TOO_LARGE Code: CL 130

Description: The real number exponent is too large.

User Action: Reduce the size of the exponent.

--ERROR CL 135-- "{text}" is not a properly formed integer.

Condition Identifier: CLE\$IMPROPER_INTEGER Code: CL 135

Description: A string assumed to represent an integer is incorrect.

User Action: Change the string to conform to the SCL syntax for integers.

--ERROR CL 140-- Improper name: {text}.

Condition Identifier: CLE\$IMPROPER_NAME Code: CL 140

Description: A string assumed to represent a name is incorrect.

User Action: Change the string to conform to the SCL syntax for names.

--ERROR CL 145-- Improper integer radix specifier: {text}.

Condition Identifier: CLE\$IMPROPER_RADIX_SPEC Code: CL 145

Description: A character other than a digit was found in the radix specification for an integer.

User Action: Change the radix to contain only digits.

--ERROR CL 150-- Improper radix value: {text}.

Condition Identifier: CLE\$IMPROPER_RADIX_VALUE Code: CL 150

Description: A radix value outside the range 2 to 16 was specified for an integer.

User Action: Choose one of the allowed radix values.

--ERROR CL 152-- "{text}" is not a properly formed real number.

Condition Identifier: CLE\$IMPROPER_REAL Code: CL 152

Description: A real number must have digits on both sides of the decimal point, and must be expressed as a decimal number. An exponent may appear immediately after the number and is separated from it by E, e, D, or d. The exponent may be signed. No spaces may appear anywhere within the number.

User Action: Correct the real number.

--ERROR CL 154-- The integer {text} is too large. Integers must be between $-(2^{63})$ and $(2^{63})-1$.

Condition Identifier: CLE\$INTEGER_LITERAL_TOO_LARGE Code: CL 154

Description: Self-explanatory.

User Action: Correct the integer.

--ERROR CL 155-- Integer value too large: {text}.

Condition Identifier: CLE\$INTEGER_TOO_LARGE Code: CL 155

Description: The specified integer exceeds the maximum value of $2^{63}-1$.

User Action: Reduce the size of the integer.

--ERROR CL 160-- Missing exponent for real number: {text}.

Condition Identifier: CLE\$MISSING_EXPONENT Code: CL 160

Description: No digits were present following the exponent designator for a real number.

User Action: Add digits to the exponent.

--ERROR CL 165-- Missing radix specifier for integer: {text}.

Condition Identifier: CLE\$MISSING_RADIX Code: CL 165

Description: The number cannot contain letters. If the number is in hexadecimal, the A, B, C, D, E, and F are interpreted as numbers; however, the number must begin with a digit and end with the radix specification (16) [for example, OFF(16)].

User Action: Either remove the letters from the number or add a hexadecimal radix specifier.

--ERROR CL 170-- Missing spaces after {text}.

Condition Identifier: CLE\$MISSING_SPACES_AFTER Code: CL 170

Description: A space is required following the stated item.

User Action: Include the space.

--ERROR CL 175-- Missing spaces before {text}.

Condition Identifier: CLE\$MISSING_SPACES_BEFORE Code: CL 175

Description: A space is required before the stated item.

User Action: Include the space.

--ERROR CL 180-- Missing string delimiter: {text}.

Condition Identifier: CLE\$MISSING_STRING_DELIMITER Code: CL 180

Description: An SCL string must be delimited by the apostrophe (') character.

User Action: Begin and terminate the string with an apostrophe.

--ERROR CL 185-- Name {text} is too long. It has more than 31 characters.

Condition Identifier: CLE\$NAME_TOO_LONG Code: CL 185

Description: An SCL name cannot be longer than 31 characters.

User Action: Shorten the name.

--ERROR CL 190-- The real number {text} is too large. The magnitude of real numbers must be between (approximately) 4.8e-1234 and 5.2e+1232.)

Condition Identifier: CLE\$REAL_LITERAL_TOO_LARGE Code: CL 190

Description: Self-explanatory.

User Action: Correct the real number.

--ERROR CL 205-- Array variable not allowed{text}.

Condition Identifier: CLE\$ARRAY_NOT_ALLOWED Code: CL 205

Description: An array variable is not allowed within the expression.

User Action: Specify a single element of the array using a subscript.

--ERROR CL 210-- Expecting value, found {text}{text}.

Condition Identifier: CLE\$EXPECTING_EXPRESSION Code: CL 210

Description: An expression cannot be empty.

User Action: Supply an expression for the value.

--ERROR CL 215-- Expecting keyword value, found {text}{text}.

Condition Identifier: CLE\$EXPECTING_KEYWORD_VALUE Code: CL 215

Description: An expression cannot be empty.

User Action: Supply and appropriate keyword value.

--ERROR CL 225-- The integer {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$INTEGER_OUT_OF_RANGE Code: CL 225

Description: The specific value is out of the valid range.

User Action: Supply an appropriate integer value.

--ERROR CL 235-- {text} not an allowed value{text}.

Condition Identifier: CLE\$NAME_NOT_A_KEYWORD_VALUE Code: CL 235

Description: The specific value is out of the valid range.

User Action: Supply an appropriate keyword value.

--ERROR CL 240-- Name {text} should not have more than {text} characters in it but it has {text}.

Condition Identifier: CLE\$NAME_VALUE_TOO_LONG Code: CL 240

Description: The specific name value is too long.

User Action: Supply a shorter name.

--ERROR CL 245-- Name {text} should have at least {text} characters in it but it has only {text}.

Condition Identifier: CLE\$NAME_VALUE_TOO_SHORT Code: CL 245

Description: The specific name value is too short.

User Action: Supply a longer name.

--ERROR CL 250-- String too long.

Condition Identifier: CLE\$STRING_TOO_LONG Code: CL 250

Description: The given string value is too long.

User Action: Supply a shorter string.

--ERROR CL 255-- String too short.

Condition Identifier: CLE\$STRING_TOO_SHORT Code: CL 255

Description: The given string value is too short.

User Action: Supply a longer string.

--ERROR CL 258-- Unable to call processor for "application value" {text}.

Condition Identifier: CLE\$UNABLE_TO_CALL_AV_SCANNER Code: CL 258

Description: The procedure that processes the stated application value could not be called. Probably it could not be loaded.

User Action: Check the residence of the processor of the application value.

--ERROR CL 260-- Unexpected array reference {text}{text}.

Condition Identifier: CLE\$UNEXPECTED_ARRAY_REFERENCE Code: CL 260

Description: Only array variables can be subscripted.

User Action: Remove the subscript from the variable reference.

--ERROR CL 265-- Wrong kind of value, expecting {text}, found {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_VALUE Code: CL 265

Description: An incorrect kind of value was supplied.

User Action: Supply the appropriate kind of value.

--ERROR CL 270-- Expecting {text} variable, found {text}{text}.

Condition Identifier: CLE\$WRONG_KIND_OF_VARIABLE Code: CL 270

Description: An incorrect kind of variable was supplied.

User Action: Supply the appropriate kind of variable.

--ERROR CL 275-- One or more of the components of a TIME_ZONE value is out of range, or the components are mutually incompatible: {text}.

Condition Identifier: CLE\$IMPOSSIBLE_TIME_ZONE Code: CL 275

Description: The time zone is not a possible entry.

User Action: Correct the time zone specification.

--ERROR CL 280-- Unrecognizable TIME_ZONE expression: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_TIME_ZONE Code: CL 280

Description: The specified time zone is not recognizable.

User Action: Correct the time zone specification.

--ERROR CL 305-- Concatenation produced too long a string.

Condition Identifier: CLE\$CONCATENATION_OVERFLOW Code: CL 305

Description: The specified concatenation operation would produce a string longer than the maximum string length.

User Action: Shorten the strings to be concatenated.

--ERROR CL 310-- Expecting end of expression, found {text}.

Condition Identifier: CLE\$EXPECTING_END_OF_EXPRESSION Code: CL 310

Description: The text was found following an expression that was apparently meant to be part of the expression but could not be interpreted as such. Probably a delimiter or operator was omitted.

User Action: Correct the expression.

--ERROR CL 315-- Expecting number or string operand, found {text} for {text} operator.

Condition Identifier: CLE\$EXPECTING_RELATION_OPERAND Code: CL 315

Description: Only numbers and strings can be compared via relational operators.

User Action: Convert the operand in question to a number or string (by using the STRING function).

--ERROR CL 320-- Expecting ')', found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN Code: CL 320

Description: A closing parenthesis character must match the opening parenthesis character within the expression.

User Action: Specify a closing parenthesis for each opening parenthesis.

--ERROR CL 325-- Division by zero is undefined.

Condition Identifier: CLE\$INTEGER_DIVIDE_BY_ZERO Code: CL 325

Description: The value following a slant (/) operator must not be zero.

User Action: Ensure that the value to the right of the slant operator cannot be zero.

--ERROR CL 330-- Integer overflow with {text} operator.

Condition Identifier: CLE\$INTEGER_OVERFLOW Code: CL 330

Description: If evaluated, the specified expression would produce an integer greater than the maximum integer value.

User Action: Correct the expression so its value cannot exceed the valid range.

--ERROR CL 335-- Operand mismatch for {text} operator.

Condition Identifier: CLE\$OPERAND_KIND_MISMATCH Code: CL 335

Description: Both operands must be of the same kind.

User Action: Change one of the operands so that its kind is compatible.

--ERROR CL 340-- Undeclared operand {text} for {text} operator.

Condition Identifier: CLE\$UNDECLARED_OPERAND Code: CL 340

Description: A name supplied for an operand is not a variable.

User Action: Either create the variable or change the operand.

--ERROR CL 345-- Expecting {text} operand, found {text} for {text} operator.

Condition Identifier: CLE\$WRONG_KIND_OF_OPERAND Code: CL 345

Description: The second operand is of the wrong kind for the operation.

User Action: Correct the operand kind.

--ERROR CL 400-- Improper access mode specification for variable {text}.

Condition Identifier: CLE\$BAD_VARIABLE_ACCESS_MODE Code: CL 400

Description:

The valid command variable access modes are clc\$read_write and clc\$read_only.

User Action:

Correct the command variable access mode specification.

--ERROR CL 401-- A bad function result was found for variable {text}.

Condition Identifier: CLE\$BAD_FUNCTION_RESULT Code: CL 401

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 402-- Bad value qualifier encountered for variable {text}.

Condition Identifier: CLE\$BAD_VALUE_QUALIFIER Code: CL 402

Description:

An invalid variable reference has been made.

User Action:

Correct the variable reference.

--ERROR CL 403-- Improper kind specification for variable {text}.

Condition Identifier: CLE\$BAD_VARIABLE_KIND Code: CL 403

Description: The valid command variable kinds are STRING, INTEGER, BOOLEAN, and STATUS.

User Action: Correct the command variable kind specification.

--ERROR CL 404-- Cannot assign a value to a read only variable: {text}.

Condition Identifier: CLE\$CANNOT_ASSIGN_TO_A_READ_VAR Code: CL 404

Description:

A variable defined with the READ attribute can only have a value assigned at the time of defining the variable. The variable cannot be modified after its definition.

User Action:

Do not change the variable value or redefine the variable without the READ attribute.

--ERROR CL 405-- Expecting field name following "." qualifier of {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_FIELD_NAME Code: CL 405

Description: When a variable reference contains a period, SCL expects a field name to follow.

User Action: Remove the period or insert a field name.

--ERROR CL 406-- Can only initialize by component an array or record in variable {text}.

Condition Identifier: CLE\$CANNOT_INITIALIZE_COMPONENT Code: CL 406

Description:

The initial value for each component of an array or record variable may be assigned individually. The initial value for other types of variables such as lists and strings must be assigned to the entire variable and not any component of the variable.

User Action:

NONE

--ERROR CL 407-- Improper scope specification for variable {text}.

Condition Identifier: CLE\$BAD_VARIABLE_SCOPE Code: CL 407

Description: The valid command variable scopes are LOCAL, JOB, XDCL, XREF, and utility name.

User Action: Correct the command variable scope specification.

--ERROR CL 408-- Cannot get the component value of variable {text} because the variable is uninitialized.

Condition Identifier: CLE\$CANNOT_READ_COMPONENT Code: CL 408

Description:

No value has been assigned to the specified component of a variable.

User Action:

Assign a value to the variable or the component if allowed.

--ERROR CL 409-- Component after field {text} is not meaningful in current value of variable {text}.

Condition Identifier: CLE\$UNDEFINED_VAR_QUALIFIER Code: CL 409

Description:

A component in the variable reference cannot be found based on the current value of the variable.

User Action:

Correct the variable reference.

--ERROR CL 410-- Expecting ')' of subscript, found {text}.

Condition Identifier: CLE\$EXPECTING_SUBSCRIPT_TERM Code: CL 410

Description: The variable reference contains an opening parenthesis character, but not a matching closing parenthesis character.

User Action: Remove the opening parenthesis character or insert a closing parenthesis character.

--ERROR CL 411-- Subscript {text} is not meaningful in current value of variable {text}.

Condition Identifier: CLE\$UNDEFINED_VAR_SUBSCRIPT Code: CL 411

Description:

The subscript in the variable reference cannot be found based on the current value of the variable.

User Action:

Correct the subscript reference.

--ERROR CL 412-- Improper string size specification for variable {text}.

Condition Identifier: CLE\$BAD_VARIABLE_STRING_SIZE Code: CL 412

Description: If the command variable kind is string, you should specify the string length.

User Action: Correct the string length specification.

--ERROR CL 413-- Substring is not meaningful in current value of variable {text}.

Condition Identifier: CLE\$UNDEFINED_VAR_SUBSTRING Code: CL 413

Description:

The substring in the variable reference cannot be found based on the current value of the variable.

User Action:

Correct the substring reference.

--ERROR CL 414-- Improper evaluation method specification for variable {text}.

Condition Identifier: CLE\$BAD_VARIABLE_EVAL_METHOD Code: CL 414

Description:

The valid command variable evaluation methods are clc\$immediate_evaluation and clc\$deferred_evaluation.

User Action:

Correct the command variable evaluation method.

--ERROR CL 415-- Expecting variable name, found {text}.

Condition Identifier: CLE\$EXPECTING_VARIABLE_NAME Code: CL 415

Description: A variable reference must begin with the variable name.

User Action: Correct the variable name.

--ERROR CL 416-- No type specification given for variable {text}.

Condition Identifier: CLE\$NO_TYPE_SPEC_SPECIFIED Code: CL 416

Description:

A type specification was not supplied for the creation of a variable.

User Action:

Supply the desired type specification when creating the variable.

--ERROR CL 417-- Cannot create variable {text} with an XREF scope and an initial value.

Condition Identifier: CLE\$XREF_VAR_CANNOT_HAVE_VALUE Code: CL 417

Description:

A variable defined with the XREF variable scope cannot have an initial value assigned in the variable definition.

User Action:

Remove the initial value in the variable definition. Modify the variable value with an assignment statement.

--ERROR CL 418-- Must specify new value in order to change the value of variable {text}.

Condition Identifier: CLE\$MUST_SPECIFY_NEW_DATA_VALUE Code: CL 418

Description:

Self-explanatory.

User Action:

Supply a new value for the variable.

--ERROR CL 419-- Cannot read an omitted parameter, {text}.

Condition Identifier: CLE\$CANNOT_READ_OMITTED_PARAM Code: CL 419

Description:

Self-explanatory.

User Action:

Ensure that an optional procedure parameter was specified before attempting to get its value.

--ERROR CL 420-- Expecting variable kind name, found {text}.

Condition Identifier: CLE\$EXPECTING_VAR_KIND_NAME Code: CL 420

Description: The valid command variable kinds are STRING, INTEGER, BOOLEAN, and STATUS.

User Action: Correct the command variable kind specification.

--ERROR CL 421-- Cannot write to an omitted parameter, {text}.

Condition Identifier: CLE\$CANNOT_WRITE_OMITTED_PARAM Code: CL 421

Description:

Self-explanatory.

User Action:

Ensure that an optional procedure parameter was specified before attempting to change its value.

--ERROR CL 422-- An internal error was encountered while trying to read variable: {text}. The type description does not match the current value.

Condition Identifier: CLE\$INTERNAL_READ_VARIABLE_ERR Code: CL 422

Description:

This is an internal error.

User Action:

Notify a site analyst to write a PSR.

--ERROR CL 423-- Cannot push variable {text} because it is not an environment variable.

Condition Identifier: CLE\$CANNOT_PUSH_VARIABLE Code: CL 423

Description:

The variable was not defined with an environment variable scope of ENVIRONMENT, JOB, TASK, UTILITY, or PUSH.

User Action:

Check the variable name and variable attributes.

--ERROR CL 425-- Expecting maximum string size, found {text}.

Condition Identifier: CLE\$EXPECTING_VAR_STRING_SIZE Code: CL 425

Description: The third parameter of a CLP\$CREATE_VARIABLE call specifies the maximum string size. If the variable kind is not string, the value is ignored, although a valid string length must be specified.

User Action: Specify a valid string length as the third parameter value.

--ERROR CL 426-- Lowerbound must be <= upperbound for variable {text}.

Condition Identifier: CLE\$IMPROPER_ARRAY_BOUNDS Code: CL 426

Description: The lowerbound variable must be less than or equal to the upperbound variable.

User Action: Decrease the lower bound value or increase the upper bound value.

--ERROR CL 427-- \$ improper first character in declaration of {text}.

Condition Identifier: CLE\$IMPROPER_USE_OF_\$ Code: CL 427

Description: The variable name cannot begin with the \$ character.

User Action: Remove the \$ from the beginning of the variable name.

--ERROR CL 428-- Improper attributes for variable {text}.

Condition Identifier: CLE\$IMPROPER_VAR_SPECIFICATION Code: CL 428

Description: The attributes (kind and dimension, for example) of a variable created with the XREF scope must exactly match those of the original creation of the variable.

User Action: Correct the variable's attributes.

--ERROR CL 429-- Improper variable name: {text}.

Condition Identifier: CLE\$IMPROPER_VARIABLE_NAME Code: CL 429

Description: The name supplied for a variable is not a valid SCL name.

User Action: Correct the name.

--ERROR CL 430-- Improper variable reference: {text}.

Condition Identifier: CLE\$IMPROPER_VARIABLE_REFERENCE Code: CL 430

Description: The value specified for a variable via CLP\$WRITE_VARIABLE is incompatible with the specified reference to that variable.

User Action: Correct either the variable reference (name) or the new value.

--ERROR CL 431-- Improper value for variable: {text}.

Condition Identifier: CLE\$IMPROPER_VARIABLE_VALUE Code: CL 431

Description: The value assigned to a variable must be of the same kind as the variable.

User Action: Specify a valid value for the variable.

--ERROR CL 442-- Wrong kind of value for variable: {text}.

Condition Identifier: CLE\$INCOMPATIBLE_ASSIGNMENT Code: CL 442

Description: The value assigned to a variable must be of the same kind as the variable.

User Action: Specify a valid value for the variable.

--ERROR CL 444-- Not enough space to hold variable {text}.

Condition Identifier: CLE\$NO_SPACE_FOR_VARIABLE Code: CL 444

Description: There is not enough space to hold the given variable.

User Action: Reduce the dimension of the variable.

--ERROR CL 445-- The initial value kind for variable {text} is not assignable to a variable.

Condition Identifier: CLE\$NOT_A_VARIABLE_KIND Code: CL 445

Description: Variables only can take values that are INTEGER, BOOLEAN, STRING or STATUS.

User Action: Use one of the allowed kinds of values.

--ERROR CL 450-- Only variable kind STRING can be qualified.

Condition Identifier: CLE\$ONLY_QUALIFY_STRING_VAR Code: CL 450

Description: The second value for the KIND parameter of CREATE_VARIABLE can be given only if the first value is STRING.

User Action: Remove the second value.

--ERROR CL 453-- {text} identifies a boolean constant and therefore may not be used as a variable name.

Condition Identifier: CLE\$SPECIAL_NAME_USE Code: CL 453

Description: A variable cannot have a name reserved for a boolean constant.

User Action: Select another name for the variable.

--ERROR CL 455-- The subscript {text} used with {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$SUBSCRIPT_OUT_OF_RANGE Code: CL 455

Description: The specified variable subscript is not within the bounds specified when the variable was defined.

User Action: Correct the subscript value.

--ERROR CL 460-- {text} is not a variable.

Condition Identifier: CLE\$UNKNOWN_VARIABLE Code: CL 460

Description: No command variable with the specified name exists within the current block.

User Action: Correct the variable name or define a variable by that name.

--ERROR CL 465-- Field {text} is not meaningful in current value of variable {text}.

Condition Identifier: CLE\$UNDEFINED_VARIABLE_FIELD Code: CL 465

Description: The only field defined for a status variable is its normal field unless the normal field value is FALSE, in which case, the identifier, condition, and text fields are also defined.

User Action: Check the normal field value before referencing other status fields.

--ERROR CL 467-- The {text} attribute is not meaningful for variable {text}.

Condition Identifier: CLE\$UNDEFINED_VAR_ATTRIBUTE Code: CL 467

Description: The specified attribute is not meaningful for this kind of variable. For example, STRING_SIZE is not meaningful for an INTEGER variable.

User Action: Check the variable's kind before interrogating kind dependent attributes.

--ERROR CL 470-- Unexpected {text} after variable reference.

Condition Identifier: CLE\$UNEXPECTED_AFTER_VARIABLE Code: CL 470

Description: Within a procedure declaration, the word VAR or ARRAY was not followed by the word OF.

User Action: Correct the declaration.

--ERROR CL 475-- Field names ({text}) are not defined for variable {text}.

Condition Identifier: CLE\$UNEXPECTED_FIELD_NAME Code: CL 475

Description: The only variable kind for which fields are defined is status.

User Action: Remove the field specification.

--ERROR CL 480-- {text} is not a field of variable {text}.

Condition Identifier: CLE\$UNKNOWN_VARIABLE_FIELD Code: CL 480

Description: A status variable has the following fields: NORMAL, IDENTIFIER, CONDITION, and TEXT.

User Action: Either remove the field identifier or specify a correct field name.

--ERROR CL 490-- {text} is already declared as a variable.

Condition Identifier: CLE\$VAR_ALREADY_CREATED Code: CL 490

Description: A variable with the specified name cannot be created because a variable having that name already exists.

User Action: Either specify another variable name or delete the existing variable.

--ERROR CL 500-- Expecting file variable or function, found {text}.

Condition Identifier: CLE\$EXPECTING_FILE_VAR_OR_FCN Code: CL 500

Description: The result of a function or variable used for the first element of a file reference must be of type FILE.

User Action: Supply a file variable or function for the first path element.

--ERROR CL 501-- The path structure is improper: {text}.

Condition Identifier: CLE\$IMPROPER_FS_PATH_STRUCTURE Code: CL 501

Description: The path string has an improper structure. It should have been obtained via pfp\$convert_string_to_fs_path.

User Action: Supply a path string obtained via pfp\$convert_string_to_fs_path.

--ERROR CL 502-- Conflicting file reference parsing options were specified.

Condition Identifier: CLE\$CONFLICTING_PARSE_OPTIONS Code: CL 502

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 503-- Catalog "{text}" is not empty.

Condition Identifier: CLE\$CATALOG_NOT_EMPTY Code: CL 503

Description: Only an empty catalog can be deleted.

User Action: Ensure that the correct catalog is specified and that all file and catalog entries have been deleted from the catalog.

--ERROR CL 504-- A file reference cannot begin with "//".

Condition Identifier: CLE\$CONCATENATE_CANT_BE_FIRST Code: CL 504

Description: A file reference cannot begin with a concatenation operator.

User Action: Correct the file reference.

--ERROR CL 505-- Cycle designator must follow file name in file reference.

Condition Identifier: CLE\$CYCLE_MUST_FOLLOW_FILE_NAME Code: CL 505

Description: The cycle reference must follow the file path.

User Action: Correct the cycle specification.

--ERROR CL 506-- Unable to find command source.

Condition Identifier: CLE\$UNABLE_TO_FIND_CMND_SOURCE Code: CL 506

Description: The residence of the requesting command or function could not be found. Perhaps \$SOURCE or \$SOURCE_OF_CALLER was used as a path element from outside of an SCL procedure or a program description.

User Action: Refer to \$SOURCE or \$SOURCE_OF_CALLER from within an SCL procedure or a program description.

--ERROR CL 507-- File variable or function value must begin with a ':'.

Condition Identifier: CLE\$MISSING_COLON_IN_VAR_OR_FCN Code: CL 507

Description: A file variable or function was incorrectly initialized. It must begin with a ':'.

User Action: Supply a correct file reference.

--ERROR CL 508-- The cycle number for file reference {file} is not known.

Condition Identifier: CLE\$CYCLE_NUMBER_NOT_KNOWN Code: CL 508

Description: The cycle number could not be returned because the information is not available to the job.

User Action: Ensure that the correct file is specified.

--ERROR CL 509-- Concatenation operator is not valid in a file variable or function value.

Condition Identifier: CLE\$NO_CONCAT_IN_VAR_OR_FCN Code: CL 509

Description: A file variable or function was incorrectly initialized. It must not contain a concatenation operator.

User Action: Supply a correct file reference.

--ERROR CL 514-- \$DEFER must be preceded by a colon.

Condition Identifier: CLE\$DEFER_MUST_FOLLOW_COLON Code: CL 514

Description: \$DEFER in a file reference must be preceded by a colon.

User Action: Correct the file reference.

--ERROR CL 520-- More than one cycle designator given in file reference.

Condition Identifier: CLE\$DUPLICATE_PF_CYCLE Code: CL 520

Description: A file reference can include only one file cycle designator.

User Action: Remove the redundant cycle designator.

--ERROR CL 525-- Expecting end of file reference, found {text}.

Condition Identifier: CLE\$EXPECTING_END_OF_FILE_REF Code: CL 525

Description: The specified text occurred immediately after a reference to a file and is not meaningful there.

User Action: Either add an appropriate delimiter or remove the text following the file reference.

--ERROR CL 530-- Expecting file reference, found {text}.

Condition Identifier: CLE\$EXPECTING_FILE_REFERENCE Code: CL 530

Description: Only a file reference is allowed.

User Action: Provide a file reference.

--ERROR CL 535-- Expecting file reference element, found {text}.

Condition Identifier: CLE\$EXPECTING_PATH_ELEMENT Code: CL 535

Description: Only a name or cycle number can follow a period (.) in a file reference.

User Action: Ensure that the components of the file reference are specified correctly.

--ERROR CL 538-- File position specifier not allowed on catalog reference.

Condition Identifier: CLE\$FILE_POSITION_NOT_ALLOWED Code: CL 538

Description: File position designators are only meaningful for sequentially accessed files; not, for example, for catalogs or library files.

User Action: Remove the file position designator.

--ERROR CL 540-- File reference too long.

Condition Identifier: CLE\$FILE_REFERENCE_TOO_LONG Code: CL 540

Description: The file reference exceeds the 256 character limit. This usually happens as a result of using a relative path to specify a file that, in conjunction with the working catalog, causes the absolute path to exceed the limit.

User Action: Either reduce the size of some of the path element names or reduce the number of path elements.

--ERROR CL 542-- {text} is only allowed on a file assigned to a device class of {text}.

Condition Identifier: CLE\$FILE_NOT_ASSIGNED_TO_DEVICE Code: CL 542

Description: This command is not allowed on the specified file because the file has not been assigned to a device class.

User Action: Use the appropriate command to assign the file to the desired device class and then retry the command.

--ERROR CL 543-- Command file qualifier {text} has improper form.

Condition Identifier: CLE\$IMPROPER_CMD_FILE_QUALIFIER Code: CL 543

Description: \$COMMAND was followed by a command file qualifier with an improper form. The command file qualifier should designate the particular source of input at the time the \$COMMAND function was called.

User Action: Use the \$COMMAND function to return a correct file reference.

--ERROR CL 544-- {text} is not allowed on a file assigned to a {text} device class.

Condition Identifier: CLE\$IMPROPER_DEVICE_CLASS Code: CL 544

Description: This command is not allowed on the specified file because the file is assigned to the wrong device class.

User Action: Use the appropriate command to assign the file to the required device class and then retry the command.

--ERROR CL 546-- {text} is not allowed on a labelled tape file.

Condition Identifier: CLE\$IMPROPER_LABELLED_TAPE_OP Code: CL 546

Description: This command is not allowed on a labelled tape file.

User Action: Detach the tape file and then specify the file label type as unlabelled before retrying the command.

--ERROR CL 548-- {text} is not allowed as an open position on the {text} command.

Condition Identifier: CLE\$IMPROPER_OPEN_POSITION Code: CL 548

Description: The open position specified on the file reference for this command is not allowed.

User Action: Change the open position on the file reference to a value that is allowed for this command.

--ERROR CL 554-- A value of {text} is improper for the {text} parameter.

Condition Identifier: CLE\$IMPROPER_VSN_VALUE Code: CL 554

Description: One or more of the characters contained in the external vsn or the recorded vsn parameter for a tape file is improper.

User Action: Change the external vsn or recorded vsn parameter value and retry the command.

--ERROR CL 555-- An invalid command file qualifier was specified: {text}.

Condition Identifier: CLE\$INVALID_CMND_FILE_QUALIFIER Code: CL 555

Description: \$COMMAND was followed by a command file qualifier which does not designate a valid source of input.

User Action: Use the \$COMMAND function to return a correct file reference.

--ERROR CL 556-- Name "{text}" is already being used as the name of a catalog.

Condition Identifier: CLE\$NAME_ALREADY_CATALOG Code: CL 556

Description: A catalog with the specified name cannot be created because a catalog having that name already exists.

User Action: Either specify another catalog name or delete the existing catalog.

--ERROR CL 558-- Name "{text}" is expected to be the name of a file but it is not.

Condition Identifier: CLE\$NAME_NOT_FILE Code: CL 558

Description: The specified name must be a file name.

User Action: Ensure that the file name is correct.

--ERROR CL 561-- The requested operation cannot be performed on the \$LOCAL catalog.

Condition Identifier: CLE\$NOT_PERMITTED_ON_LOC_CAT Code: CL 561

Description: The catalog must be a permanent file catalog.

User Action: Specify a permanent file catalog.

--ERROR CL 562-- The requested operation cannot be performed on a file in the \$LOCAL catalog.

Condition Identifier: CLE\$NOT_PERMITTED_ON_LOC_FILE Code: CL 562

Description: The file must be a permanent file.

User Action: Specify a permanent file catalog and a file registered in the catalog.

--ERROR CL 563-- The requested operation can be performed only on a temporary file in the \$LOCAL catalog.

Condition Identifier: CLE\$ONLY_PERMITTED_ON_LOC_FILE Code: CL 563

Description: The requested operation can be performed only on a temporary file in catalog \$LOCAL.

User Action: Use a temporary file.

--ERROR CL 564-- Only cycle 1 is allowed for file {file}.

Condition Identifier: CLE\$ONLY_CYCLE_ONE_ALLOWED Code: CL 564

Description: Only cycle one is allowed.

User Action: No user action required.

--ERROR CL 565-- Position designator not allowed on command or entry_point reference.

Condition Identifier: CLE\$NO_POS_ON_CMND_ENTRY_PT_REF Code: CL 565

Description: A file position (i.e. \$ASIS, \$BOI, or \$EOI) is not allowed on a command reference or on an entry point reference.

User Action: Correct the command or entry_point reference.

--ERROR CL 567-- Position designator must follow file name in file reference.

Condition Identifier: CLE\$POSITION_MUST_BE_LAST Code: CL 567

Description: A period follows the position designator.

User Action: Remove the period following the position designator.

--ERROR CL 568-- Use of command file reference (i.e. \$COMMAND or \$COMMAND_OF_CALLER) is inappropriate for the requested operation.

Condition Identifier: CLE\$INAPPROPRIATE_CMND_FILE_REF Code: CL 568

Description: Self-explanatory.

User Action: Use a specific file for the request.

--ERROR CL 570-- {text} must be first element of file reference.

Condition Identifier: CLE\$SPECIAL_ELEMENT_NOT_FIRST Code: CL 570

Description: The specified element is reserved for use at the beginning of a file reference.

User Action: Use a name other than the reserved ones.

--ERROR CL 575-- System error - see job log.

Condition Identifier: CLE\$SYSTEM_ERROR Code: CL 575

Description: An internal error has occurred.

User Action: See the job log.

--ERROR CL 578-- Expecting file reference, found \$SYSTEM command source.

Condition Identifier: CLE\$UNEXPECTED_SYS_CMND_SOURCE Code: CL 578

Description: When used as a generic path name, \$SOURCE/\$SOURCE_OF_CALLER must refer to an object library or catalog. It refers to \$SYSTEM.

User Action: Refer to \$SOURCE/\$SOURCE_OF_CALLER from within an SCL procedure or a program description.

--ERROR CL 581-- Expecting file reference, found {text} utility command source.

Condition Identifier: CLE\$UNEXPECTED_UTIL_CMND_SOURCE Code: CL 581

Description: When used as a generic path name, \$SOURCE/\$SOURCE_OF_CALLER must refer to an object library or catalog. It refers to a utility.

User Action: Refer to \$SOURCE/\$SOURCE_OF_CALLER from within an SCL procedure or a program description.

--ERROR CL 585-- A ":" may only be used at the beginning of a path.

Condition Identifier: CLE\$UNEXPECTED_COLON_IN_PATH Code: CL 585

Description: In a path name for a file or catalog, a colon may appear only before the family name. Colons may not be used between elements of a path name.

User Action: Use periods to separate elements of a path name.

--ERROR CL 591-- "/" must be followed by a name or integer variable or function: {text} was specified.

Condition Identifier: CLE\$VAR_OR_FCN_FOLLOWS_CONCAT Code: CL 591

Description: The concatenation operator in a file reference must be followed by a name or integer variable or function.

User Action: Correct the file reference.

--ERROR CL 592-- :\$DEFER must be followed by a file variable or function: {text} was specified.

Condition Identifier: CLE\$VAR_OR_FCN_FOLLOWS_\$DEFER Code: CL 592

Description: \$DEFER in a file reference must be followed by a file variable or function.

User Action: Correct the file reference.

--ERROR CL 605-- Improperly constructed parameter description table.

Condition Identifier: CLE\$BAD_PDT Code: CL 605

Description: Self-explanatory.

User Action: Use the GENERATE_PDT command to produce a correct version of the parameter description table (PDT).

--ERROR CL 607-- A clt\$parameter_value_table is not the same size as its associated clt\$parameter_description_table.

Condition Identifier: CLE\$BAD_PVT Code: CL 607

Description: Self-explanatory.

User Action: Correct the declaration of the parameter value table (PVT) variable. If possible, use the GENERATE_PDT command which produces the declaration of the PVT as well as the PDT.

--ERROR CL 610-- Parameter {text} already given.

Condition Identifier: CLE\$DOUBLY_DEFINED_PARAMETER Code: CL 610

Description: The same parameter is specified more than once in the parameter list.

User Action: Remove the redundant parameter specification.

--ERROR CL 615-- Expecting end of default value expression for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_DEFAULT_TERM Code: CL 615

Description: The specified text occurred immediately following the default specification and is not meaningful there. Probably a delimiter was omitted.

User Action: Correct the default specification.

--ERROR CL 620-- Expecting end of parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAMETER_TERM Code: CL 620

Description: The specified text occurred immediately following the default specification and is not meaningful there. Probably a delimiter was omitted.

User Action: Correct the parameter specification.

--ERROR CL 623-- Expecting ")" of parameter list for function {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_PLIST Code: CL 623

Description: Rparen of plist expected.

User Action: Correct the parameter list.

--ERROR CL 625-- Expecting ')' of value list for parameter {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_VLIST Code: CL 625

Description: A closing parenthesis does not follow the number of values expected in the value list.

User Action: Specify a closing parenthesis immediately after the last value in the value list.

--ERROR CL 630-- Expecting ')' of value set for parameter {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_VSET Code: CL 630

Description: A closing parenthesis does not follow the number of values expected in the value list.

User Action: Specify a closing parenthesis immediately after the last value in the value set.

--ERROR CL 635-- Expecting end of value for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_VALUE_ELEM_TERM Code: CL 635

Description: The specified text occurred immediately following a value for the parameter and is not meaningful there. Probably a delimiter was omitted.

User Action: Correct the value specification.

--ERROR CL 640-- Expecting end of value set for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_VALUE_SET_TERM Code: CL 640

Description: The specified text occurred immediately following a value set for the parameter and is not meaningful there. Probably a delimiter was omitted.

User Action: Correct the value specification.

--ERROR CL 645-- The parameter_list passed to clp\$scan_parameter_list is garbled.

Condition Identifier: CLE\$GARBLED_PARAMETER_LIST Code: CL 645

Description: The parameter list is not in a form recognizable by SCL.

User Action: Check that a valid parameter list is passed by the call.

--INFORMATIVE CL 646-- The parameters for a command or function were successfully displayed. This message should never appear to a user; if it does, the processor for the command or function whose parameters were displayed is not interfacing properly with the SCL interpreter.

Condition Identifier: CLE\$PARAMETERS_DISPLAYED Code: CL 646

Description: The parameters for a command were successfully displayed.

User Action: Don't use the DISPLAY_COMMAND_INFORMATION command with the command that caused this message to appear.

--ERROR CL 650-- Parameter {text} is required but was omitted.

Condition Identifier: CLE\$REQUIRED_PARAMETER_OMITTED Code: CL 650

Description: One of the parameters required for command processing has not been specified in the command parameter list.

User Action: Specify all required parameters. If necessary, refer to the command description.

--ERROR CL 653-- Semicolon cannot be used as separator when replying to prompting.

Condition Identifier: CLE\$SEMICOLON_NOT_SEPARATOR Code: CL 653

Description: A semicolon may only be used at the end of a reply to an SCL parameter prompt.

User Action: Correct the use of the semicolon.

--ERROR CL 655-- Too few values given for parameter {text}.

Condition Identifier: CLE\$TOO_FEW_VALUES Code: CL 655

Description: Too few values were given for the specific parameter.

User Action: Provide at least the minimum number of values required for the parameter.

--ERROR CL 660-- Too few values in value set given for parameter {text}.

Condition Identifier: CLE\$TOO_FEW_VALUES_IN_SET Code: CL 660

Description: Too few values in the value set were given for the specific parameter.

User Action: Provide at least the minimum number of values required for each value set for the parameter.

--ERROR CL 665-- Too few value sets given for parameter {text}.

Condition Identifier: CLE\$TOO_FEW_VALUE_SETS Code: CL 665

Description: Too few value sets were given for the specific parameter.

User Action: Provide at least the minimum number of value sets required for the parameter.

--ERROR CL 670-- Too many parameters given.

Condition Identifier: CLE\$TOO_MANY_PARAMETERS Code: CL 670

Description: Too many parameters were given. This usually occurs when parameters are given positionally or when a command has no parameters.

User Action: Either specify the parameters by name or remove the excess parameters.

--ERROR CL 675-- Too many values given for parameter {text}.

Condition Identifier: CLE\$TOO_MANY_VALUES Code: CL 675

Description: Too many values were given for the specific parameter

User Action: Provide no more than the maximum number of values allowed for the parameter.

--ERROR CL 680-- Too many values in value set given for parameter {text}.

Condition Identifier: CLE\$TOO_MANY_VALUES_IN_SET Code: CL 680

Description: Too many values in the value set were given for the specific parameter.

User Action: Provide no more than the maximum number of values allowed for each value set for the parameter.

--ERROR CL 685-- Too many value sets given for parameter {text}.

Condition Identifier: CLE\$TOO_MANY_VALUE_SETS Code: CL 685

Description: Too many value sets were given for the specific parameter.

User Action: Provide no more than the maximum number of value sets allowed for the parameter.

--ERROR CL 686-- Unable to call procedure to do command/function specific parameter checking for clp\$evaluate_parameters.

Condition Identifier: CLE\$UNABLE_TO_CALL_CHECK_PROC Code: CL 686

Description: The specified CLT\$CHECK_PARAMETERS_PROCEDURE could not be called. Probably it could not be loaded.

User Action: Check the residence of the specified procedure.

--ERROR CL 688-- Unexpected {text} in parameter list.

Condition Identifier: CLE\$UNEXPECTED_IN_PARAM_LIST Code: CL 688

Description: A ";" cannot be used as a separator within the parameter list for a function, nor can a ")" be used to terminate the parameter list of a command.

User Action: Correct the parameter list.

--ERROR CL 690-- {text} is not a parameter name.

Condition Identifier: CLE\$UNKNOWN_PARAMETER_NAME Code: CL 690

Description: The specified parameter name is not defined in the parameter descriptor table for the command.

User Action: Correct the parameter name.

--ERROR CL 692-- "VAR (or ARRAY) OF type" is not supported in combination with LIST, RANGE, or KEY specifications (parameter {text}).

Condition Identifier: CLE\$UNSUPPORTED_PARAMETER_SPEC Code: CL 692

Description: The parameter specification supplied is unsupported.

User Action: Correct the parameter specification.

--ERROR CL 694-- A parameter was given positionally but there is no parameter defined for that position; value = {text}.

Condition Identifier: CLE\$VALUE_GIVEN_POSITIONALLY Code: CL 694

Description: A command parameter value was entered for a parameter that is not defined for the command.

User Action: Correct the parameter specification.

--ERROR CL 695-- Value range not allowed for parameter {text}.

Condition Identifier: CLE\$VALUE_RANGE_NOT_ALLOWED Code: CL 695

Description: The value range is not allowed for the specific parameters.

User Action: Remove the range specification for the parameter value.

--ERROR CL 705-- Assignment statements cannot be labelled or preceded by '/.

Condition Identifier: CLE\$ASSIGNMENT_CANT_BE_LABELLED Code: CL 705

Description: The assignment statement has a label or reverse slant () character before it. Only structured statements can be labeled.

User Action: Move the label so that it is not immediately before an assignment statement.

--ERROR CL 707-- Login command is either missing or incorrect in {file}.

Condition Identifier: CLE\$BAD_OR_MISSING_LOGIN_IN_JOB Code: CL 707

Description: Either the LOGIN command is missing or its parameter values are incorrect (for example, the password may be incorrect) or the file submitted is empty (it exists but has SIZE=0).

User Action: Ensure that the submit file is not empty and that it contains a valid LOGIN command.

--FATAL CL 708-- Command aborted: {text}

Condition Identifier: CLE\$COMMAND_ABORTED Code: CL 708

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 709-- Command file is not accessible.

Condition Identifier: CLE\$COMMAND_FILE_NOT_ACCESSIBLE Code: CL 709

Description: The environment in which a file referred to originally as \$COMMAND or \$COMMAND_OF_CALLER is no longer present, thereby preventing access to the intended command stream.

User Action: Respecify the intended source of commands as \$COMMAND or \$COMMAND_OF_CALLER or other file as appropriate.

--ERROR CL 710-- Command/procedure file {file} is not callable.

Condition Identifier: CLE\$COMMAND_FILE_NOT_CALLABLE Code: CL 710

Description: To execute the contents of the file, it must be accessed from within its execute ring bracket. The ring at which the file is called must be greater than or equal to the file's R1 ring attribute and less than or equal to its R2 ring attribute.

User Action: Request the file owner to change the file's ring attributes (CHANGE_FILE_ATTRIBUTE command) to allow execution of the file.

--ERROR CL 711-- Command/procedure file {file} is not executable.

Condition Identifier: CLE\$COMMAND_FILE_NOT_EXECUTABLE Code: CL 711

Description: To execute the contents of the file, you must have execute access to the file.

User Action: Request that the file owner change the applicable access control entry to permit execute permission.

--ERROR CL 714-- Commands cannot be labelled.

Condition Identifier: CLE\$COMMANDS_CANT_BE_LABELLED Code: CL 714

Description: Only structured statements can be labeled.

User Action: Move the label so that it is not immediately before a command.

--FATAL CL 715-- Command search aborted due to system condition or segment access condition on file {file}.

Condition Identifier: CLE\$COMMAND_SEARCH_ABORTED Code: CL 715

Description: While a library was being searched for a specified command, a system condition or segment access condition was encountered.

User Action: Check that the file is valid, i.e. that it is open, etc.

--ERROR CL 716-- Continued command line too long.

Condition Identifier: CLE\$CONTINUED_LINE_TOO_LONG Code: CL 716

Description: The combined length of the initial line and its continuation lines cannot exceed 65535 characters.

User Action: Shorten the input line.

--ERROR CL 717-- Attempt to add {file} to the command list, but it is already in the list.

Condition Identifier: CLE\$DUPLICATE_COMMAND_LIST_ENT Code: CL 717

Description: The file is already in the command list.

User Action: None.

--ERROR CL 718-- Attempt to add \$SYSTEM to the command list, but it is already in the list.

Condition Identifier: CLE\$DUPLICATE_\$SYSTEM_ENTRY Code: CL 718

Description: Self-explanatory.

User Action: No action required.

--ERROR CL 719-- '/' not allowed before command in exclusive mode.

Condition Identifier: CLE\$ESCAPE_NOT_ALLOWED Code: CL 719

Description: Because the current command list search mode is exclusive, preceding a command with the escape character ? is invalid.

User Action: Either change the search mode or enter only commands that are valid in exclusive search mode.

--ERROR CL 720-- "Escaped" command not allowed: {text}.

Condition Identifier: CLE\$ESCAPED_COMMAND_NOT_ALLOWED Code: CL 720

Description: A command may not be prefixed with a "/" if the command search made is exclusive.

User Action: Don't use the "/" prefix.

--ERROR CL 725-- Expecting command, found {text}.

Condition Identifier: CLE\$EXPECTING_COMMAND Code: CL 725

Description: The SCL interpreter could not recognize the line as a command. Probably a delimiter is missing.

User Action: Correct the command specification.

--ERROR CL 729-- Expecting command reference, found {text}.

Condition Identifier: CLE\$EXPECTING_COMMAND_REFERENCE Code: CL 729

Description: A command reference was expected.

User Action: Supply a command reference.

--ERROR CL 730-- Expecting continuation of command line, found end of input.

Condition Identifier: CLE\$EXPECTING_CONTINUATION_LINE Code: CL 730

Description: Although the previous line ended with an ellipsis (..), the next line contained no text, only the end-of-information.

User Action: Either remove the ellipsis on the previous line or complete the command with a continuation line.

--ERROR CL 732-- Expecting entry point reference, found {text}.

Condition Identifier: CLE\$EXPECTING_ENTRY_POINT_REF Code: CL 732

Description: An entry point reference was expected.

User Action: Supply an entry point reference.

--ERROR CL 735-- File.command ({text}) not allowed in "exclusive" command mode.

Condition Identifier: CLE\$FILE_DOT_CMND_NOT_ALLOWED Code: CL 735

Description: Because the current command list search mode is exclusive, the file command syntax cannot be used.

User Action: Enter only commands that are valid in exclusive command search mode.

--ERROR CL 740-- File {text} not found in input stack.

Condition Identifier: CLE\$FILE_NOT_IN_INPUT_STACK Code: CL 740

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 744-- Attributes of command file {file} are improper: {text}.

Condition Identifier: CLE\$IMPROPER_COMMAND_FILE_ATTR Code: CL 744

Description: The specified file attributes are improper for the intended use of the file.

User Action: Use a file with the appropriate file attributes.

--ERROR CL 748-- Line from command file {file} is too long.

Condition Identifier: CLE\$LINE_TOO_LONG Code: CL 748

Description: A line from a command file has more than 256 characters in it.

User Action: Shorten the line.

--ERROR CL 749-- The input to INCLUDE_COMMAND consisted of more than one command: {text}

Condition Identifier: CLE\$MULTIPLE_INCLUDE_COMMAND Code: CL 749

Description: Only one command may be specified on each call to CLP\$INCLUDE_COMMAND or the INCLUDE_COMMAND command.

User Action: If more than one command is to be executed, either use CLP\$SCAN_COMMAND_LINE (or the INCLUDE_LINE command), or make one call to CLP\$INCLUDE_COMMAND (or the INCLUDE_COMMAND command) for each command to be executed.

--ERROR CL 750-- File {file} is not a command file.

Condition Identifier: CLE\$NOT_A_COMMAND_FILE Code: CL 750

Description: A file containing commands must have the following file attribute values: FILE_CONTENTS of LEGIBLE (or UNKNOWN), FILE_PROCESSOR of SCL (or UNKNOWN), and FILE_STRUCTURE of DATA (or UNKNOWN).

User Action: Use only a file with the appropriate attributes.

--ERROR CL 751-- File {file} can not be used as an EXECUTE_COMMAND command file.

Condition Identifier: CLE\$INCORRECT_EXEC_COMMAND_FILE Code: CL 751

Description: Self-explanatory.

User Action: Use COLLECT_TEXT or some other method to create a file of commands to be passed to EXECUTE_COMMAND.

--ERROR CL 752-- The input to EXECUTE_COMMAND consisted of more than one command: {text}

Condition Identifier: CLE\$MULTIPLE_EXECUTE_COMMAND Code: CL 752

Description: Only one command may be specified on each call to CLA\$EXECUTE_COMMAND or the EXECUTE_COMMAND.

User Action: Use only one call for each command to be executed.

--ERROR CL 753-- A {text} statement can not be used as an EXECUTE_COMMAND command.

Condition Identifier: CLE\$INVALID_EXEC_COMMAND Code: CL 753

Description: Self-explanatory.

User Action: No action required.

--ERROR CL 754-- {text} may not be used as a EXECUTE_COMMAND task name.

Condition Identifier: CLE\$INVALID_EXEC_TASK_NAME Code: CL 754

Description: Self-explanatory.

User Action: Choose a different name for the task.

--WARNING CL 755-- {file} not in job command list.

Condition Identifier: CLE\$NOT_IN_COMMAND_LIST Code: CL 755

Description: The given file is not in the job command list.

User Action: None.

--ERROR CL 757-- Search mode cannot be changed with an active utility and a restrictive search mode.

Condition Identifier: CLE\$CANNOT_CHANGE_SEARCH_MODE Code: CL 757

Description: The search mode cannot be changed while the current search mode is restrictive and a utility is being executed.

User Action: Exit the current utility before attempting to change the command search mode.

--ERROR CL 759-- The command list and command search mode cannot be changed while the command search mode is exclusive.

Condition Identifier: CLE\$EXCLUSVE_MODE_EXCLUDES_CMND Code: CL 759

Description: The exclusive mode excludes command.

User Action: No action required.

--ERROR CL 761-- Cannot {text} entry at front of command list.

Condition Identifier: CLE\$RESTRICTED_MODE_CMND_CHANGE Code: CL 761

Description: The front of the command list cannot be changed while the search mode is restrictive.

User Action: Change the search mode to global.

--ERROR CL 775-- Unable to call command: {text}.

Condition Identifier: CLE\$UNABLE_TO_CALL_COMMAND Code: CL 775

Description: The processor for a command (usually a utility subcommand) could not be called by the SCL interpreter.

User Action: Contact the supplier of the utility.

--ERROR CL 780-- Unbalanced block structure: found {text}, expecting {text}.

Condition Identifier: CLE\$UNBALANCED_BLOCK_STRUCTURE Code: CL 780

Description: A structured statement was not terminated when the end of a command file or procedure was encountered.

User Action: Complete the structured statement.

--ERROR CL 785-- Unexpected {text} after command.

Condition Identifier: CLE\$UNEXPECTED_AFTER_COMMAND Code: CL 785

Description: The specified text immediately followed a command name. Probably a delimiter was omitted.

User Action: Correct the command call.

--ERROR CL 786-- Unexpected ',' after {text}.

Condition Identifier: CLE\$UNEXPECTED_COMMA_AFTER Code: CL 786

Description: A comma cannot follow the name of a control statement.

User Action: Use a space rather than a comma.

--ERROR CL 788-- Unexpected '/' before {text}.

Condition Identifier: CLE\$UNEXPECTED_ESCAPE Code: CL 788

Description: Control statements cannot be preceded by the reverse slant (?) character.

User Action: Remove the reverse slant character from the control statement.

--ERROR CL 789-- Unexpected '?' before {text}.

Condition Identifier: CLE\$UNEXPECTED_PROMPT_STATEMENT Code: CL 789

Description: The "prompt for parameters" notation cannot be used with a control statement, it applies only to commands.

User Action: Remove the "?" preceding the control statement.

--ERROR CL 790-- {text} is not a command.

Condition Identifier: CLE\$UNKNOWN_COMMAND Code: CL 790

Description: The name does not designate a command in the current command list as controlled by the command search mode.

User Action: Ensure that the command is spelled correctly.

--ERROR CL 805-- Duplicated keyword value: {text}.

Condition Identifier: CLE\$DUPLICATE_KEYWORD_VALUE Code: CL 805

Description: The declaration specified the same keyword more than once for a parameter.

User Action: Remove the redundant keyword specification.

--ERROR CL 810-- Duplicate parameter name {text} in {text} declaration.

Condition Identifier: CLE\$DUPLICATE_PARAMETER_NAME Code: CL 810

Description: The declaration specified the same parameter name more than once.

User Action: Remove the redundant parameter name specification.

--ERROR CL 815-- Duplicate name {text} in {text} declaration.

Condition Identifier: CLE\$DUPLICATE_PROC_NAME Code: CL 815

Description: The declaration specified the same procedure name more than once.

User Action: Remove the redundant procedure name specification.

--ERROR CL 820-- Expecting name for application value scanner, found {text}.

Condition Identifier: CLE\$EXPECTING_AVIS_NAME Code: CL 820

Description: Within a declaration, the only valid specification after the value kind name for an application value is the name of the application value scanner.

User Action: Correct the declaration.

--ERROR CL 825-- Expecting ';' or END OF LINE, found {text}.

Condition Identifier: CLE\$EXPECTING_COMMAND_SEPARATOR Code: CL 825

Description: Only a command separator can follow the closing parenthesis of a procedure declaration.

User Action: Add a semicolon (;) after the closing parenthesis (]) of the procedure declaration or move the text that follows it to the next line.

--ERROR CL 830-- Expecting KEY, found {text}.

Condition Identifier: CLE\$EXPECTING_KEY_SPEC Code: CL 830

Description: Within a procedure declaration, the word OR following a value kind specification was not followed by a keyword specification.

User Action: Correct the declaration.

--ERROR CL 835-- Expecting parameter definition, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAM_DEF Code: CL 835

Description: Within a procedure declaration, instead of the specification for a parameter, the stated text was found.

User Action: Correct the declaration.

--ERROR CL 840-- Expecting ';' or END OF LINE, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAM_DEF_SEP Code: CL 840

Description: Within a procedure declaration, the specification for a parameter was not followed by a semicolon or was not at the end of a line.

User Action: Correct the declaration.

--ERROR CL 845-- Expecting parameter name, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAM_NAME Code: CL 845

Description: Within a procedure declaration, the stated text was found rather than a parameter name.

User Action: Correct the declaration.

--ERROR CL 850-- Expecting {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_PROC Code: CL 850

Description: A procedure declaration did not begin with the word PROC (or a parameter descriptor table declaration did not begin with the word PDT).

User Action: Correct the declaration.

--ERROR CL 855-- Expecting {text} name in declaration, found {text}.

Condition Identifier: CLE\$EXPECTING_PROC_NAME Code: CL 855

Description: Within a procedure declaration, instead of a procedure name, the stated text was found.

User Action: Correct the declaration.

--ERROR CL 860-- Expecting value kind name, found {text}.

Condition Identifier: CLE\$EXPECTING_VALUE_KIND Code: CL 860

Description: Within a procedure declaration, instead of the name of a value kind the stated text was found.

User Action: Correct the declaration.

--ERROR CL 865-- Expecting variable kind name, found {text}.

Condition Identifier: CLE\$EXPECTING_VAR_KIND Code: CL 865

Description: Within a procedure declaration, instead of the name of a variable kind, the stated text was found.

User Action: Correct the declaration.

--ERROR CL 870-- Uppervalue for {text} omitted.

Condition Identifier: CLE\$INTEGER_UPPERVALUE_OMITTED Code: CL 870

Description: Within a procedure declaration, the qualifier for an INTEGER value kind was not given as range.

User Action: Correct the declaration.

--ERROR CL 875-- Low greater than high for {text}.

Condition Identifier: CLE\$LOW_GREATER_THAN_HIGH Code: CL 875

Description: Within a procedure declaration, the low value of the range for the qualifier of a value kind was greater than the corresponding high value.

User Action: Correct the declaration.

--ERROR CL 880-- Too many parameter definitions.

Condition Identifier: CLE\$TOO_MANY_PARAMETER_DEFS Code: CL 880

Description: Within a procedure declaration, too many parameters were defined.

User Action: Remove the excess parameters.

--ERROR CL 885-- Too many parameter names.

Condition Identifier: CLE\$TOO_MANY_PARAMETER_NAMES Code: CL 885

Description: Within a procedure declaration, too many procedure names were defined.

User Action: Remove the excess procedure names.

--ERROR CL 890-- Only one name allowed for PDT.

Condition Identifier: CLE\$TOO_MANY_NAMES_FOR_PDT Code: CL 890

Description: Within the declaration for a parameter descriptor table (PDT), only one name can be given to the PDT.

User Action: Remove the excess PDT names.

--ERROR CL 900-- Unexpected {text} after INTEGER.

Condition Identifier: CLE\$UNEXPECTED_AFTER_INTEGER Code: CL 900

Description: Within a procedure declaration, the value kind name INTEGER was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 905-- Unexpected {text} after KEY.

Condition Identifier: CLE\$UNEXPECTED_AFTER_KEY Code: CL 905

Description: Within a procedure declaration, the value kind name KEY was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 910-- Unexpected {text} after LIST.

Condition Identifier: CLE\$UNEXPECTED_AFTER_LIST Code: CL 910

Description: Within a procedure declaration, the word LIST was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 915-- Unexpected {text} after NAME.

Condition Identifier: CLE\$UNEXPECTED_AFTER_NAME Code: CL 915

Description: Within a procedure declaration, the value kind name NAME was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 920-- Unexpected {text} after OF.

Condition Identifier: CLE\$UNEXPECTED_AFTER_OF Code: CL 920

Description: Within a procedure declaration, the word OF was followed by invalid text.

User Action: Correct the statement.

--ERROR CL 925-- Unexpected {text} after parameter name.

Condition Identifier: CLE\$UNEXPECTED_AFTER_PARAM_NAME Code: CL 925

Description: Within a procedure declaration, a parameter name was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 930-- Unexpected {text} after {text}.

Condition Identifier: CLE\$UNEXPECTED_AFTER_PROC Code: CL 930

Description: Within a procedure declaration, invalid text followed the word PROC (or, within a parameter descriptor table declaration, invalid text followed the word PDT).

User Action: Correct the declaration.

--ERROR CL 935-- Unexpected {text} after proc name.

Condition Identifier: CLE\$UNEXPECTED_AFTER_PROC_NAME Code: CL 935

Description: Within a procedure declaration, a parameter name was followed by invalid text.

User Action: Correct the statement.

--ERROR CL 940-- Unexpected {text} after RANGE.

Condition Identifier: CLE\$UNEXPECTED_AFTER_RANGE Code: CL 940

Description: Within a procedure declaration, the word RANGE was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 945-- Unexpected {text} after STRING.

Condition Identifier: CLE\$UNEXPECTED_AFTER_STRING Code: CL 945

Description: Within a procedure declaration, the value kind name STRING was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 950-- Unexpected {text} after value kind.

Condition Identifier: CLE\$UNEXPECTED_AFTER_VALUE_KIND Code: CL 950

Description: Within a procedure declaration, a value kind specification was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 955-- Unexpected {text} after VAR.

Condition Identifier: CLE\$UNEXPECTED_AFTER_VAR Code: CL 955

Description: Invalid text occurred immediately after a reference to a variable and is not meaningful there.

Probably a delimiter is needed to separate the variable reference from what follows.

User Action: Either add an appropriate delimiter or remove the text following the variable reference.

--ERROR CL 960-- Unexpected {text} after value count.

Condition Identifier: CLE\$UNEXPECTED_AFTER_VC_SPEC Code: CL 960

Description: Within a procedure declaration, a value count specification was followed by invalid text.

User Action: Correct the statement.

--ERROR CL 965-- Unexpected {text} after value set count.

Condition Identifier: CLE\$UNEXPECTED_AFTER_VSC_SPEC Code: CL 965

Description: Within a procedure declaration, a value set count specification was followed by invalid text.

User Action: Correct the declaration.

--ERROR CL 970-- Unexpected OF after ', '.

Condition Identifier: CLE\$UNEXPECTED_OF_AFTER_COMMA Code: CL 970

Description: Within a procedure declaration, a value list specification was followed by the word OF.

User Action: Correct the declaration.

--ERROR CL 1005-- Improper argument descriptor table for function {text} : {text}.

Condition Identifier: CLE\$BAD_ADT Code: CL 1005

Description: The ADT parameter must be a pointer to a variable of type CLT\$ARGUMENT_DESCRIPTOR_TABLE. The ADT is an adaptable array of argument descriptor records.

User Action: Correct the ADT variable specification, declaration, or initialization.

--ERROR CL 1020-- Expecting end of argument {text} of function {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_ARGUMENT_TERM Code: CL 1020

Description: The valid argument terminators are a comma (,) or a space.

User Action: Terminate the argument appropriately.

--ERROR CL 1030-- Expecting ')' of argument list for function {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_ALIST Code: CL 1030

Description: A closing parenthesis character must terminate the argument list.

User Action: Specify a closing parenthesis after the argument list.

--ERROR CL 1060-- Argument {text} is required but was omitted from function {text}.

Condition Identifier: CLE\$REQUIRED_ARGUMENT_OMITTED Code: CL 1060

Description: One of the arguments required for function processing has not been specified.

User Action: Specify all required arguments. If necessary, refer to the function description.

--ERROR CL 1070-- Too many arguments given for function {text}.

Condition Identifier: CLE\$TOO_MANY_ARGUMENTS Code: CL 1070

Description: More arguments than necessary were specified in the function argument list.

User Action: Remove the unneeded arguments.

--ERROR CL 1080-- Unable to call function: {text}.

Condition Identifier: CLE\$UNABLE_TO_CALL_FUNCTION Code: CL 1080

Description: The processor for the function (usually a utility supplied function) could not be called by the SCL interpreter.

User Action: Contact the supplier of the utility.

--ERROR CL 1090-- {text} is not a function.

Condition Identifier: CLE\$UNKNOWN_FUNCTION Code: CL 1090

Description: The specified function name is not defined within this block.

User Action: Check that the function name is correct.

--ERROR CL 1105-- Cannot pop command list because a utility was added.

Condition Identifier: CLE\$CANNOT_POP_COMMAND_LIST Code: CL 1105

Description: Cannot pop the command list while in the utility.

User Action: Exit the utility before attempting to pop the command list.

--ERROR CL 1106-- CASE selection statement cannot be labelled.

Condition Identifier: CLE\$CASE_SEL_CANT_BE_LABELLED Code: CL 1106

Description: Only the structured statements BLOCK, FOR, LOOP, REPEAT, and WHILE can be labelled.

User Action: Remove the label.

--ERROR CL 1107-- WHEN clause found more than once in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$DUPLICATE_WHEN_CLAUSE Code: CL 1107

Description: A duplicate WHEN clause occurred.

User Action: Correct the statement.

--ERROR CL 1108-- WITH clause found more than once in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$DUPLICATE_WITH_CLAUSE Code: CL 1108

Description: A duplicate WITH clause occurred.

User Action: Correct the statement.

--ERROR CL 1110-- Expecting '=', found {text} after FOR statement control variable.

Condition Identifier: CLE\$EXPECTING_FOR_ASSIGN Code: CL 1110

Description: An incorrect symbol was found after the FOR statement control variable.

User Action: Correct the FOR statement.

--ERROR CL 1111-- Expecting BY or DO, found {text} in FOR statement.

Condition Identifier: CLE\$EXPECTING_FOR_BY_OR_DO Code: CL 1111

Description: Only DO or BY can follow the final value specification in a FOR statement.

User Action: Correct the FOR statement.

--ERROR CL 1112-- Expecting TO, found {text} in FOR statement.

Condition Identifier: CLE\$EXPECTING_FOR_TO Code: CL 1112

Description: Only TO can follow the initial value specification in a FOR statement.

User Action: Correct the FOR statement.

--ERROR CL 1113-- Expecting IN, found {text} in FOR statement.

Condition Identifier: CLE\$EXPECTING_FOR_IN Code: CL 1113

Description: Only IN can follow the control variable of a FOR EACH statement.

User Action: Correct the FOR statement.

--ERROR CL 1114-- Expecting variable reference or EACH, found {text} in FOR statement.

Condition Identifier: CLE\$EXPECTING_FOR_VAR_OR_EACH Code: CL 1114

Description: The first item in a FOR statement must be the control variable for the statement or keyword EACH.

User Action: Correct the FOR statement.

--ERROR CL 1115-- Expecting variable reference, found {text} in FOR statement.

Condition Identifier: CLE\$EXPECTING_FOR_VARIABLE Code: CL 1115

Description: The control variable is the first parameter of the FOR or FOR EACH statement.

User Action: Correct the FOR statement.

--ERROR CL 1120-- Expecting label, found {text} in {text} statement.

Condition Identifier: CLE\$EXPECTING_LABEL Code: CL 1120

Description: Only a label is allowed in this position.

User Action: Correct the statement.

--ERROR CL 1122-- Expecting label or WHEN, found {text} in CYCLE statement.

Condition Identifier: CLE\$EXPECTING_LABEL_OR_WHEN Code: CL 1122

Description: Only a label or WHEN statement is allowed in this position.

User Action: Correct the statement.

--ERROR CL 1123-- Expecting label, WHEN or WITH, found {text} in EXIT statement.

Condition Identifier: CLE\$EXPECTING_LABEL_WHEN_WITH Code: CL 1123

Description: Expecting a label.

User Action: Correct the statement.

--ERROR CL 1126-- Expecting RETRY or WHEN, found {text} in CONTINUE statement.

Condition Identifier: CLE\$EXPECTING_RETRY_OR_WHEN Code: CL 1126

Description: Only RETRY or WHEN is allowed in a CONTINUE statement.

User Action: Correct the statement.

--ERROR CL 1127-- Expecting WHEN, found {text} in CONTINUE statement.

Condition Identifier: CLE\$EXPECTING_CONTINUE_WHEN Code: CL 1127

Description: Expecting CONTINUE_WHEN.

User Action: Correct the statement.

--ERROR CL 1128-- Expecting WHEN, found {text} in CYCLE statement.

Condition Identifier: CLE\$EXPECTING_CYCLE_WHEN Code: CL 1128

Description: Expecting CYCLE_WHEN.

User Action: Correct the statement.

--ERROR CL 1129-- Expecting WHEN, found {text} in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$EXPECTING_EXIT_WHEN Code: CL 1129

Description: Expecting EXIT_WHEN.

User Action: Correct the statement.

--ERROR CL 1130-- Expecting WITH, found {text} in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$EXPECTING_WITH Code: CL 1130

Description: Self-explanatory.

User Action: Correct the EXIT statement.

--ERROR CL 1131-- Expecting WITH or WHEN, found {text} in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$EXPECTING_WITH_OR_WHEN Code: CL 1131

Description: Only WITH or WHEN is allowed in the EXIT_PROC statement.

User Action: Correct the EXIT_PROC statement.

--ERROR CL 1132-- Improper name for environment object or variable: {text}.

Condition Identifier: CLE\$IMPROPER_ENV_OBJECT_NAME Code: CL 1132

Description: The name given for the environment object is not a valid SCL name.

User Action: Correct the environment object name.

--ERROR CL 1133-- {text} cannot be popped.

Condition Identifier: CLE\$NO_OBJECT_TO_POP Code: CL 1133

Description: There is no previous version of the environment object to restore.

User Action: None.

--ERROR CL 1134-- {text} is not an environment object or variable.

Condition Identifier: CLE\$NOT_AN_ENVIRONMENT_OBJECT Code: CL 1134

Description: The specified name does not designate an environment object.

User Action: Supply a valid environment object name.

--ERROR CL 1135-- {text} has already been pushed at this level.

Condition Identifier: CLE\$OBJECT_ALREADY_PUSHED Code: CL 1135

Description: A previous version of the environment object has been saved.

User Action: Pop the environment object. Note that this action restores the environment object to its previous version.

--ERROR CL 1138-- {text} statement cannot be cycled.

Condition Identifier: CLE\$STATEMENT_CANT_BE_CYCLED Code: CL 1138

Description: The CYCLE statement only can refer to an interactive structured statement (for example, FOR, LOOP, REPEAT or WHILE).

User Action: Correct the statement.

--ERROR CL 1140-- {text} statement cannot be labelled.

Condition Identifier: CLE\$STATEMENT_CANT_BE_LABELLED Code: CL 1140

Description: Only the structured statements BLOCK, FOR, LOOP, REPEAT, and WHILE can be labeled.

User Action: Remove the label.

--ERROR CL 1150-- Unexpected {text} after boolean expression.

Condition Identifier: CLE\$UNEXPECTED_AFTER_BOOL_EXPR Code: CL 1150

Description: The stated text followed the boolean expression on a control statement.

User Action: Correct the statement.

--ERROR CL 1152-- Unexpected {text} after condition name in {text} statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_COND_NAME Code: CL 1152

Description: The entry after the condition name was incorrect.

User Action: Correct the statement.

--ERROR CL 1155-- Unexpected {text} after label in {text} statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_END_LABEL Code: CL 1155

Description: The entry after label is incorrect.

User Action: Remove the text that follows the label.

--ERROR CL 1160-- Unexpected {text} after BY in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_BY Code: CL 1160

Description: The entry after BY in the FOR statement is incorrect.

User Action: Correct the FOR statement.

--ERROR CL 1161-- Unexpected {text} after final value in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_FINAL Code: CL 1161

Description: The entry after the final value in the FOR statement is incorrect.

User Action: Correct the FOR statement.

--ERROR CL 1162-- Unexpected {text} after initial value in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_INIT Code: CL 1162

Description: The entry after the initial value in the FOR statement is incorrect.

User Action: Correct the FOR statement.

--ERROR CL 1163-- Unexpected {text} after step value in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_STEP Code: CL 1163

Description: The entry after the step value in the FOR statement is incorrect.

User Action: Correct the FOR statement.

--ERROR CL 1164-- Unexpected {text} after TO in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_TO Code: CL 1164

Description: The entry after TO in the FOR statement is incorrect.

User Action: Correct the FOR statement.

--ERROR CL 1165-- Unexpected {text} after list value in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_LIST Code: CL 1165

Description: Only DO may follow the list of the FOR EACH statement.

User Action: Correct the FOR statement.

--ERROR CL 1166-- Unexpected {text} after environment object name.

Condition Identifier: CLE\$UNEXPECTED_AFTER_OBJ_NAME Code: CL 1166

Description: Only a comma, space, semicolon, or end of line may follow an environment object name in a PUSH or POP statement.

User Action: Correct the statement.

--ERROR CL 1167-- Unexpected {text} after IN in FOR statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_FOR_IN Code: CL 1167

Description: Only a list expression may follow IN in the FOR EACH statement.

User Action: Correct the FOR statement.

--ERROR CL 1168-- Unexpected {text} after RETRY in CONTINUE statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_RETRY Code: CL 1168

Description: The entry after RETRY in the CONTINUE statement is incorrect.

User Action: Correct the CONTINUE statement.

--ERROR CL 1170-- Unexpected {text} after {text}.

Condition Identifier: CLE\$UNEXPECTED_AFTER_THEN_OR_DO Code: CL 1170

Description: The entry after THEN or DO is incorrect.

User Action: Remove the text after THEN or DO, or move it to the next line, or add a semicolon (;) after THEN or DO.

--ERROR CL 1173-- Unexpected {text} after WHEN in {text} statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_WHEN Code: CL 1173

Description: The entry after WHEN in the specified statement is incorrect.

User Action: Correct the statement.

--ERROR CL 1174-- Unexpected {text} after WITH in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_WITH Code: CL 1174

Description: The entry after WITH in the EXIT_PROC statement is incorrect.

User Action: Correct the EXIT_PROC statement.

--ERROR CL 1175-- Unexpected {text} after WITH value in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_WITH_VALUE Code: CL 1175

Description: The entry after the WITH value in the EXIT_PROC statement is incorrect.

User Action: Correct the EXIT_PROC statement.

--ERROR CL 1176-- Unexpected {text} after WHEN value in EXIT or EXIT_PROC statement.

Condition Identifier: CLE\$UNEXPECTED_AFTER_WHEN_VALUE Code: CL 1176

Description: Self-explanatory.

User Action: Correct the EXIT statement.

--ERROR CL 1180-- Unexpected {text} statement.

Condition Identifier: CLE\$UNEXPECTED_CONTROL_STATEMENT Code: CL 1180

Description: The specified statement part must match with a corresponding structured statement initial clause (for example, an IFEND must have a corresponding IF statement).

User Action: Correct the structured statement usage.

--ERROR CL 1184-- Unexpected FUNCTION declaration.

Condition Identifier: CLE\$UNEXPECTED_FUNCTION Code: CL 1184

Description: Functions cannot be declared dynamically.

User Action: Put the function on an object library.

--ERROR CL 1185-- Unexpected PROCEDURE declaration.

Condition Identifier: CLE\$UNEXPECTED_PROC Code: CL 1185

Description: Procedures can not be declared dynamically.

User Action: Put the procedure onto a file or object library.

--ERROR CL 1190-- Unexpected parameters for {text} statement.

Condition Identifier: CLE\$UNEXPECTED_STATEMENT_PARAMS Code: CL 1190

Description: The specified statement has no parameters.

User Action: Remove the parameters.

--ERROR CL 1193-- WITH clause only allowed in an EXIT statement that designates a PROC.

Condition Identifier: CLE\$WITH_CLAUSE_NOT_ALLOWED Code: CL 1193

Description: A WITH clause is not allowed.

User Action: Correct the statement.

--ERROR CL 1195-- Wrong statement label for {text} statement.

Condition Identifier: CLE\$WRONG_STATEMENT_LABEL Code: CL 1195

Description: The label specified on the terminator of a structured statement does not match the label specified at the beginning of the statement.

User Action: Correct the label specification.

--ERROR CL 1260-- {text} designates a task which has not yet completed.

Condition Identifier: CLE\$TASK_NAME_IN_USE Code: CL 1260

Description: The specified name cannot be used for an asynchronous task until the task by that name that is still executing terminates.

User Action: Choose a different task name.

--ERROR CL 1280-- You may not use TASK/TASKEND to run below your validated minimum ring.

Condition Identifier: CLE\$TASK_TASKEND_RING_BELOW_MIN Code: CL 1280

Description: TASK/TASKEND cannot be used to run below a validated minimum ring.

User Action: None.

--ERROR CL 1310-- Attempt to make circular file connection from {file} to {file}.

Condition Identifier: CLE\$CIRCULAR_FILE_CONNECTION Code: CL 1310

Description: The files are already connected. For example, if file A is already connected to file B, an attempt to connect file B to file A would fail.

User Action: If appropriate, you could delete the previous connection.

--ERROR CL 1315-- Connection between {file} and {file} may not be broken.

Condition Identifier: CLE\$CONNECTION_CANNOT_BE_BROKEN Code: CL 1315

Description: The specified file connection cannot be deleted.

User Action: Ensure that the correct files are specified on the deletion request.

--WARNING CL 1320-- Connection between {file} and {file} already exists.

Condition Identifier: CLE\$DUPLICATE_FILE_CONNECTION Code: CL 1320

Description: Only one connection can exist between two files at a time.

User Action: None.

--ERROR CL 1325-- Cannot connect {file} and {file} because of incompatible attributes: {text}.

Condition Identifier: CLE\$INCOMPATIBLE_FILE_CONNECT Code: CL 1325

Description: The FILE_CONTENTS and FILE_STRUCTURE attributes of files connected to each other must be equal. An attribute value of UNKNOWN (the default) is considered equal to any other value.

User Action: Change the file attributes of one of the files or use different files for the connection.

--ERROR CL 1330-- Improper name for subject of file connection: {text}.

Condition Identifier: CLE\$IMPROPER_SUBJECT_FILE_NAME Code: CL 1330

Description: The name given for the the subject file parameter on the CLE\$CREATE_FILE_CONNECTION request was not a valid SCL name.

User Action: Use a correct name.

--ERROR CL 1335-- Improper name for target of file connection: {text}.

Condition Identifier: CLE\$IMPROPER_TARGET_FILE_NAME Code: CL 1335

Description: The name given for the target file parameter on the CLP\$CREATE_FILE_CONNECTION request was not a valid SCL name.

User Action: Use a correct name.

--ERROR CL 1370-- {file} may not be made the subject of a file connection because it already exists.

Condition Identifier: CLE\$SUBJECT_CANNOT_BE_CONNECTED Code: CL 1370

Description: An existing file that is not a connected file cannot become a connected file.

User Action: Use a different file.

--WARNING CL 1380-- {file} is not connected to {file}.

Condition Identifier: CLE\$UNKNOWN_FILE_CONNECTION Code: CL 1380

Description: The connection between the two files cannot be deleted because it does not exist.

User Action: None.

--ERROR CL 1405-- Ending) is missing from language string.

Condition Identifier: CLE\$LANGUAGE_DELIMITER_MISSING Code: CL 1405

Description: The natural language must be surrounded by parentheses.

User Action: Add a parenthesis.

--ERROR CL 1407-- Cannot find message module for specified language {text}.

Condition Identifier: CLE\$LANGUAGE_MODULE_NOT_FOUND Code: CL 1407

Description: A version of the specified message module could not be found for the specified natural language.

User Action: Use a different natural language.

--ERROR CL 1408-- Template for {text} is invalid for {text}.

Condition Identifier: CLE\$BAD_TEMPLATE_FOR_MONTH_DAY Code: CL 1408

Description: The message template for the name of a month or day is not in the proper format.

User Action: Correct the month or day name template.

--ERROR CL 1410-- Date_time format may not be null or blank.

Condition Identifier: CLE\$DATE_TIME_FORMAT_NULL Code: CL 1410

Description: Neither a date, time, or format string have been supplied.

User Action: Supply a format string, or a date-time string.

--ERROR CL 1415-- {text} is not allowed in this date_time format context.

Condition Identifier: CLE\$UNEXPECTED_DT_FORMAT_CHAR Code: CL 1415

Description: A character other than those accepted as standard input has been found in the context of a date or time or date-time format.

User Action: Be sure that all characters supplied are correct in the context of the date or format.

--ERROR CL 1420-- Specified date_time string is not a known date_time format.

Condition Identifier: CLE\$UNKNOWN_DATE_TIME_FORMAT Code: CL 1420

Description: Part of the specified date_time does not match the format that was supplied, or if no format was supplied, a format cannot be deduced based on the given date_time string.

User Action: Correct the date-time and/or format input.

--ERROR CL 1425-- {text} is not a known month or day name.

Condition Identifier: CLE\$NAME_NOT_MONTH_OR_DAY Code: CL 1425

Description: A name supplied for a month or day is not one of the standard months of the year or days of the week.

User Action: Correct the name.

--ERROR CL 1430-- One or more of the components of a DATE_TIME value is out of range, or the components are mutually incompatible: {text}.

Condition Identifier: CLE\$IMPOSSIBLE_DATE_OR_TIME Code: CL 1430

Description: An integer supplied for a time or date component is out of range, or the combination of components designates a date that cannot occur.

User Action: Correct the date-time input.

--ERROR CL 1435-- Too many components in date_time string {text}.

Condition Identifier: CLE\$DATE_TIME_STRING_TOO_LONG Code: CL 1435

Description: Self-explanatory.

User Action: Shorten the input data.

--ERROR CL 1440-- Incorrect month number specified.

Condition Identifier: CLE\$BAD_MONTH_NUMBER Code: CL 1440

Description: The number of a month must be between 1 and 12.

User Action: Correct the month number.

--ERROR CL 1502-- Utility command list entries cannot be moved in the command list.

Condition Identifier: CLE\$CANNOT_MOVE_UTILITY_ENTRY Code: CL 1502

Description: The utility entry cannot be moved.

User Action: No action required.

--ERROR CL 1504-- Command source is not a library.

Condition Identifier: CLE\$COMMAND_SOURCE_NOT_LIB Code: CL 1504

Description:

The file containing the processors for a command utility's commands and functions is not a library.

User Action:

Correct the library parameter on the UTILITY command.

--ERROR CL 1506-- Command source is unknown.

Condition Identifier: CLE\$COMMAND_SOURCE_UNKNOWN Code: CL 1506

Description: This is an internal error.

User Action: Notify a site analyst to write a PSR.

--ERROR CL 1508-- Utility attributes had key(s) that do not apply to this request in array element(s): {text}.

Condition Identifier: CLE\$IMPROPER_UTILITY_ATTRIBUTE Code: CL 1508

Description: The utility attribute specified is improper.

User Action: Correct the utility attribute keys.

--ERROR CL 1510-- Utility attributes had improper value(s) in array element(s): {text}.

Condition Identifier: CLE\$IMPROPER_UTILITY_ATTR_VALUE Code: CL 1510

Description: The value of the UTILITY attribute is improper.

User Action: Correct the specified UTILITY attribute values.

--ERROR CL 1512-- {text} may not be used as a utility name.

Condition Identifier: CLE\$IMPROPER_UTILITY_NAME Code: CL 1512

Description: Only valid SCL names can be used as utility names.

User Action: Correct the utility name.

--WARNING CL 1514-- Interactive include processor for utility {text} is currently active. The processor is ignored.

Condition Identifier: CLE\$INCLUDE_PROCESSOR_ACTIVE Code: CL 1514

Description: Include processor active.

User Action: No action required.

--ERROR CL 1515-- No utility is currently active.

Condition Identifier: CLE\$NO_UTILITY_ACTIVE Code: CL 1515

Description:

There are no active utilities to associate a variable defined with the variable scope of UTILITY.

User Action:

Define the variable with a variable scope other than UTILITY.

--ERROR CL 1516-- Termination command {text} not defined in command table.

Condition Identifier: CLE\$TERM_COMMAND_NOT_DEFINED Code: CL 1516

Description: The termination command for a command level utility was not defined in the utility's command table.

User Action: Define the termination command in the command table.

--ERROR CL 1518-- Unable to call interactive include processor: {text}.

Condition Identifier: CLE\$UNABLE_TO_CALL_INC_PROCESOR Code: CL 1518

Description: The interactive include processor for a utility could not be called by the SCL interpreter.

User Action: Contact the supplier of the utility.

--ERROR CL 1520-- Unable to call line preprocessor: {text}.

Condition Identifier: CLE\$UNABLE_TO_CALL_PREPROCESSOR Code: CL 1520

Description: The line preprocessor for a utility could not be called by the SCL interpreter.

User Action: Contact the supplier of the utility.

--ERROR CL 1522-- Utility attributes had unknown key(s) in array element(s): {text}.

Condition Identifier: CLE\$UNKNOWN_UTILITY_ATTRIBUTE Code: CL 1522

Description: The utility attribute is unknown.

User Action: Correct the specified utility attribute keys.

--ERROR CL 1524-- Utility command/function {text} is not available.

Condition Identifier: CLE\$UTIL_CMDS_FCTNS_UNAVAILABLE Code: CL 1524

Description: The specified name was found to be a utility command or function. But the command/function cannot be executed because the current task is different from the utility task.

User Action: Return to the utility task.

--WARNING CL 1526-- Utility commands/functions left in the command list.

Condition Identifier: CLE\$UTILITY_LEFT_IN_CMND_LIST Code: CL 1526

Description: The utility was left in the command list.

User Action: No action required.

--ERROR CL 1528-- {text} does not designate a utility that is currently active.

Condition Identifier: CLE\$UNKNOWN_UTILITY Code: CL 1528

Description: A reference was made to a utility that is not known to SCL.

User Action: Correct the utility name.

--ERROR CL 1530-- The utility {text} cannot be accessed by {text}.

Condition Identifier: CLE\$INACCESSIBLE_UTILITY Code: CL 1530

Description:

Change_utility_attributes command can only reference utilities defined by UTILITY/UTILITYEND.

Cip\$change_utility_attributes can only reference utilities defined by the clp\$begin_utility and that the utility is available in the current task.

User Action:

Ensure that the correct method is used to change a utility's attributes.

--ERROR CL 2000-- Expecting attribute for APPLICATION type, found {text}.

Condition Identifier: CLE\$EXPECTING_APPLIC_TYPE_ATTR Code: CL 2000

Description: Self-explanatory.

User Action: Correct the APPLICATION type declaration.

--ERROR CL 2001-- {text} elements given for array, but the array bounds are {text} .. {text}.

Condition Identifier: CLE\$TOO_FEW_OR_MANY_ARRAY_ELEMS Code: CL 2001

Description: Self-explanatory.

User Action: Supply the correct number of elements for the array.

--ERROR CL 2002-- The ARRAY bound {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$ARRAY_BOUND_OUT_OF_RANGE Code: CL 2002

Description: Self-explanatory.

User Action: Correct the array bounds declaration.

--ERROR CL 2003-- In the declaration for an ARRAY type, the bounds must be specified unless the declaration is for the type of a command (procedure) or function parameter.

Condition Identifier: CLE\$ARRAY_BOUNDS_REQUIRED Code: CL 2003

Description: Self-explanatory.

User Action: Correct the array declaration.

--ERROR CL 2004-- In the declaration for an ARRAY type, the element type must be specified unless the declaration is for the type of a command (procedure) or function parameter.

Condition Identifier: CLE\$ARRAY_ELEM_TYPE_REQUIRED Code: CL 2004

Description: Self-explanatory.

User Action: Correct the array declaration.

--ERROR CL 2005-- Expecting OF in ARRAY type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_OF_FOR_ARRAY Code: CL 2005

Description: Self-explanatory.

User Action: Correct the array declaration.

--ERROR CL 2006-- The maximum bound for an array was omitted but must be given when the minimum bound is given.

Condition Identifier: CLE\$MAX_ARRAY_BOUND_OMITTED Code: CL 2006

Description: Self-explanatory.

User Action: Correct the array declaration.

--ERROR CL 2007-- The minimum bound for an array may not be greater than the corresponding maximum bound.

Condition Identifier: CLE\$MIN_ARRAY_BOUND_GT_MAX Code: CL 2007

Description: Self-explanatory.

User Action: Correct the array declaration.

--ERROR CL 2008-- The operands of the "AND" operator must be booleans, a {text} was found.

Condition Identifier: CLE\$AND_OPERAND_NOT_BOOLEAN Code: CL 2008

Description: Self-explanatory.

User Action: Correct the operand.

--ERROR CL 2009-- The operand of the "NOT" operator must be a boolean, a {text} was found.

Condition Identifier: CLE\$NOT_OPERAND_NOT_BOOLEAN Code: CL 2009

Description: Self-explanatory.

User Action: Correct the operand.

--ERROR CL 2010-- The operands of the "OR" operator must be booleans, a {text} was found.

Condition Identifier: CLE\$OR_OPERAND_NOT_BOOLEAN Code: CL 2010

Description: Self-explanatory.

User Action: Correct the operand.

--ERROR CL 2011-- The operands of the "XOR" operator must be booleans, a {text} was found.

Condition Identifier: CLE\$XOR_OPERAND_NOT_BOOLEAN Code: CL 2011

Description: Self-explanatory.

User Action: Correct the operand.

--ERROR CL 2012-- Expecting COBOL_name expression, found {text}.

Condition Identifier: CLE\$EXPECTING_COBOL_NAME_EXPR Code: CL 2012

Description: Self-explanatory.

User Action: Supply a name that conforms to the syntax for a name in the COBOL programming language.

--ERROR CL 2013-- {text} is not a COBOL_name. A COBOL name may contain up to 30 letters, digits or hyphens. It may not start or end with a hyphen and must contain at least one letter.

Condition Identifier: CLE\$NOT_A_COBOL_NAME Code: CL 2013

Description: Self-explanatory.

User Action: Supply a name that conforms to the syntax for a name in the COBOL programming language.

--ERROR CL 2014-- Values of type {text} may not be compared.

Condition Identifier: CLE\$NON_COMPARABLE_TYPE Code: CL 2014

Description: The relational operators may not be used to compare values of the named type.

User Action: None.

--ERROR CL 2015-- Values of types {text} and {text} may not be compared.

Condition Identifier: CLE\$NON_COMPARABLE_VALUES Code: CL 2015

Description: The relational operators may not be used to compare values of the named types.

User Action: Use a function to convert one of the operands to the type required.

--ERROR CL 2016-- Expecting DATE_TIME expression, found {text}.

Condition Identifier: CLE\$EXPECTING_DATE_TIME_EXPR Code: CL 2016

Description: Self-explanatory.

User Action: Supply a date_time value.

--ERROR CL 2017-- Function {text} can only be read; it may not be used as a variable or parameter.

Condition Identifier: CLE\$FUNCTION_IS_READ_ONLY Code: CL 2017

Description: Only the functions \$VARIABLE and \$PARAMETER_VALUE can be assigned to.

User Action: Use a variable or parameter reference.

--ERROR CL 2018-- Unrecognizable DATE_TIME expression: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_DATE_TIME Code: CL 2018

Description: Self-explanatory.

User Action: Correct the date_time value.

--ERROR CL 2019-- The tense for {text} "{text}" ({text}) is wrong, it must be: {text}.

Condition Identifier: CLE\$WRONG_DATE_TIME_TENSE Code: CL 2019

Description: Self-explanatory.

User Action: Change the date_time to one with an appropriate tense.

--ERROR CL 2020-- Expecting tense (PAST, PRESENT or FUTURE) in DATE/TIME type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_DATE_TIME_TENSE Code: CL 2020

Description: Self-explanatory.

User Action: Correct the date_time type declaration.

--ERROR CL 2021-- Expecting semicolon or end of line after VAR definition, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_VAR_DEF Code: CL 2021

Description: Self-explanatory.

User Action: Correct the variable declaration.

--ERROR CL 2022-- Field {text} cannot be accessed because of the values of other fields within {text}.

Condition Identifier: CLE\$UNACCESSIBLE_FIELD Code: CL 2022

Description:

Self-explanatory.

User Action:

Correct the field reference.

--ERROR CL 2023-- Field {text} within {text} has not been assigned a value.

Condition Identifier: CLE\$UNDEFINED_FIELD Code: CL 2023

Description:

Self-explanatory.

User Action:

Correct the field reference.

--ERROR CL 2024-- {text} is not a field within {text}.

Condition Identifier: CLE\$UNKNOWN_FIELD Code: CL 2024

Description: The name is not a keyword for the field.

User Action: Specify a correct keyword.

--ERROR CL 2025-- Expecting colon after VAR name, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_VAR_NAME Code: CL 2025

Description: Self-explanatory.

User Action: Correct the variable declaration.

--ERROR CL 2026-- No result was returned by function {text}.

Condition Identifier: CLE\$NO_FUNCTION_RESULT Code: CL 2026

Description: A function processor must either set its result parameter to point to a clt\$data_value or return an abnormal status.

User Action: Correct the function processor.

--ERROR CL 2027-- Variable, {text}, defined with the read attribute must be given an initial value.

Condition Identifier: CLE\$READ_VAR_REQUIRES_VALUE Code: CL 2027

Description:

A variable defined with the READ attribute must have an initial value assigned at the time of the variable definition.

User Action:

Supply an initial value or redefine the variable without the READ attribute.

--ERROR CL 2028-- The radix {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$RADIX_OUT_OF_RANGE Code: CL 2028

Description: Self-explanatory.

User Action: Correct the radix for the integer type.

--ERROR CL 2029-- {text} is not a keyword.

Condition Identifier: CLE\$UNKNOWN_KEYWORD Code: CL 2029

Description: The name is not a keyword for the parameter or variable.

User Action: Specify a correct keyword.

--ERROR CL 2030-- Duplicate keyword {text} in {text} declaration.

Condition Identifier: CLE\$DUPLICATE_KEYWORD Code: CL 2030

Description: Within a KEY type declaration a particular keyword may only appear once.

User Action: Correct the KEY type declaration.

--ERROR CL 2031-- Expecting keyword in KEY type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_KEY_IN_SPEC Code: CL 2031

Description: Self-explanatory.

User Action: Correct the KEY type declaration.

--ERROR CL 2032-- No keywords following ADVANCED_KEY in KEY type specification.

Condition Identifier: CLE\$NO_ADVANCED_USAGE_KEYWORDS Code: CL 2032

Description: There must be at least one keyword defined following ADVANCED_KEY in a KEY type declaration.

User Action: Correct the KEY type declaration.

--ERROR CL 2033-- No keywords following HIDDEN_KEY in KEY type specification.

Condition Identifier: CLE\$NO_HIDDEN_KEYWORDS Code: CL 2033

Description: There must be at least one keyword defined following HIDDEN_KEY in a KEY type declaration.

User Action: Correct the KEY type declaration.

--ERROR CL 2034-- No keywords in KEY type specification.

Condition Identifier: CLE\$NO_KEYWORDS Code: CL 2034

Description: There must be at least one keyword defined following KEY in a KEY type declaration.

User Action: Correct the KEY type declaration.

--ERROR CL 2035-- No keywords before ADVANCED_KEY or HIDDEN_KEY in KEY type specification.

Condition Identifier: CLE\$NO_NORMAL_USAGE_KEYWORDS Code: CL 2035

Description: There must be at least one keyword defined following KEY in a KEY type declaration.

User Action: Correct the KEY type declaration.

--ERROR CL 2036-- The modification name {text} is too long. It has more than 9 characters.

Condition Identifier: CLE\$MODIFICATION_NAME_TOO_LONG Code: CL 2036

Description: The modification_name field of the built in record type line_identifier may not exceed nine characters in length.

User Action: Correct the modification_name.

--ERROR CL 2037-- The line_identifier's sequence number must be an integer, not a real number, but {text} was given.

Condition Identifier: CLE\$SEQUENCE_NUMBER_NOT_INTEGER Code: CL 2037

Description: Self-explanatory.

User Action: Correct the sequence_number.

--ERROR CL 2038-- The line_identifier's sequence number must be between 1 and 16777215, but {text} was given.

Condition Identifier: CLE\$SEQUENCE_NUM_OUT_OF_RANGE Code: CL 2038

Description: Self-explanatory.

User Action: Correct the sequence_number.

--ERROR CL 2039-- Expecting space, ',' or ')' after list element, found {text}.

Condition Identifier: CLE\$EXPECTING_LIST_ELEM_SEP Code: CL 2039

Description: Self-explanatory.

User Action: Correct the list expression.

--ERROR CL 2040-- Expecting ")" of list, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_LIST Code: CL 2040

Description: Self-explanatory.

User Action: Add a right parenthesis to the end of the list expression.

--ERROR CL 2041-- {text} elements given for list, but the number of elements must be between {text} and {text}.

Condition Identifier: CLE\$TOO_FEW_OR_MANY_LIST_ELEMS Code: CL 2041

Description: Self-explanatory.

User Action: Supply the correct number of list elements.

--ERROR CL 2042-- Expecting OF in LIST type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_OF_FOR_LIST Code: CL 2042

Description: Self-explanatory.

User Action: Correct the list type declaration.

--ERROR CL 2043-- The REST qualifier for a LIST type may only be specified for the last field of a RECORD, or the last parameter of a function (provided the parameter does not have the VAR attribute).

Condition Identifier: CLE\$IMPROPER_USE_OF_LIST_REST Code: CL 2043

Description: Self-explanatory.

User Action: Remove the REST attribute from the list declaration.

--ERROR CL 2045-- In the declaration for a LIST type, the element type must be specified unless the declaration is for the type of a command (procedure) or function parameter.

Condition Identifier: CLE\$LIST_ELEM_TYPE_REQUIRED Code: CL 2045

Description: Self-explanatory.

User Action: Correct the list declaration.

--ERROR CL 2046-- The maximum bound for a list was omitted but must be given when the minimum bound is given.

Condition Identifier: CLE\$MAX_LIST_BOUND_OMITTED Code: CL 2046

Description: Self-explanatory.

User Action: Correct the list declaration.

--ERROR CL 2047-- The minimum bound for a list may not be greater than the corresponding maximum bound.

Condition Identifier: CLE\$MIN_LIST_BOUND_GT_MAX Code: CL 2047

Description: Self-explanatory.

User Action: Correct the list declaration.

--ERROR CL 2048-- Expecting type expression, found {text}.

Condition Identifier: CLE\$EXPECTING_TYPE_EXPRESSION Code: CL 2048

Description: The type declaration for parameter variable or type must begin with the name of one of the standard types, or a variable whose value is a type specification or a function that returns a type specification.

User Action: Correct the type declaration.

--ERROR CL 2049-- Duplicate attribute {text} for variable {text}.

Condition Identifier: CLE\$DUPLICATE_VARIABLE_ATTR Code: CL 2049

Description: Within a variable declaration a particular attribute may appear only once.

User Action: Correct the variable declaration.

--ERROR CL 2050-- Expecting operand of expression, found {text}.

Condition Identifier: CLE\$EXPECTING_OPERAND Code: CL 2050

Description: An expression or operand of an expression may not begin with the text named in the message.

User Action: Correct the expression or operand.

--ERROR CL 2051-- Expecting ")" of subexpression, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_SUBEXPR Code: CL 2051

Description: A sub-expression must be enclosed in parentheses.

User Action: Add a ')' at the end of the sub-expression.

--ERROR CL 2052-- Wrong kind of value, value out of range, or improper expression for a union (ANY) type: {text}.

Condition Identifier: CLE\$EXPRESSION_NOT_UNION_TYPE Code: CL 2052

Description: The expression supplied does not conform to any of the types allowed.

User Action: Correct the expression. If you can't determine what specifically is wrong, create a variable of the type you are trying to specify and assign the expression to that variable. A message will be produced describing the error in greater detail.

--ERROR CL 2053-- Unable to evaluate deferred value for {text}.

Condition Identifier: CLE\$NONEVALUABLE_DEFERRED_VALUE Code: CL 2053

Description: Self-explanatory.

User Action: Correct the value.

--ERROR CL 2054-- A reference to unspecified (omitted) parameter {text} may not be the operand of the "{text}" operator.

Condition Identifier: CLE\$UNEXPECTED_OPER_FOR_UNSPEC Code: CL 2054

Description: An omitted parameter may only be used in an expression when it is the only item in the expression.

User Action: Correct the expression.

--ERROR CL 2055-- A reference to unspecified (omitted) parameter {text} may not be followed by a subscript, substring, field reference or path element; a "{text}" was found.

Condition Identifier: CLE\$UNEXPECTED_QUAL_FOR_UNSPEC Code: CL 2055

Description: Self-explanatory.

User Action: Correct the reference.

--WARNING CL 2056-- The {text} request resulted in an unspecified value (omitted parameter {text}).

Condition Identifier: CLE\$UNSPECIFIED_VALUE_FOR_REQ Code: CL 2056

Description: Self-explanatory.

User Action: Ensure that an optional procedure parameter was specified before referencing it.

--FATAL CL 2058-- Attempt to use unrecognizable value with the "{text}" operator.

Condition Identifier: CLE\$UNRECOGNIZABLE_DATA_VALUE Code: CL 2058

Description: All of the necessary fields of a CLT\$DATA_VALUE record have not been correctly "filled in", probably by a function processor.

User Action: Make sure that all relevant fields of the record are set to an appropriate value.

--ERROR CL 2059-- Wrong kind of list or array element type, expecting {text}, found {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_ELEMENT_TYPE Code: CL 2059

Description: The type specification of a list or array passed to a variable must match the type specification defined for that variable.

User Action: Supply the correct element type for the list or array.

--ERROR CL 2060-- Wrong kind of list or array element value, expecting {text}, found {text} for element number {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_ELEMENT_VALUE Code: CL 2060

Description: Self-explanatory.

User Action: Make sure that all elements of the array or list are of the correct type.

--ERROR CL 2061-- Expecting attribute for variable {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_VARIABLE_ATTR Code: CL 2061

Description: Self-explanatory.

User Action: Correct the variable declaration.

--ERROR CL 2062-- "{text}" is not a variable attribute.

Condition Identifier: CLE\$NOT_A_VARIABLE_ATTRIBUTE Code: CL 2062

Description: Self-explanatory.

User Action: Correct the variable declaration.

--ERROR CL 2063-- When the \$VALUE function is used to refer to a VAR parameter only the parameter name may be given.

Condition Identifier: CLE\$QUAL_VALUE_IS_READ_ONLY Code: CL 2063

Description: Self-explanatory.

User Action: Remove the extra parameters from the \$VALUE call or, better, simply refer to the procedure parameter by its name, or use the \$PARAMETER_VALUE function.

--ERROR CL 2064-- The minimum size for a name may not be greater than the corresponding maximum size.

Condition Identifier: CLE\$MIN_NAME_SIZE_GT_MAX Code: CL 2064

Description: Self-explanatory.

User Action: Correct the name declaration.

--ERROR CL 2065-- The name size {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$NAME_SIZE_OUT_OF_RANGE Code: CL 2065

Description: Self-explanatory.

User Action: Correct the name declaration.

--ERROR CL 2066-- The operands of the "{text}" operator must be numbers (integer or real), a {text} was found.

Condition Identifier: CLE\$ARITHMETIC_OPERAND_NOT_NUM Code: CL 2066

Description: Self-explanatory.

User Action: Correct the expression.

--ERROR CL 2067-- Arithmetic overflow occurred attempting {text}.

Condition Identifier: CLE\$ARITHMETIC_OVERFLOW Code: CL 2067

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2068-- A loss of significant digits resulted from attempting {text}.

Condition Identifier: CLE\$ARITHMETIC_SIGNIFICANCE Code: CL 2068

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2069-- A divide fault (division by zero) occurred attempting {text}.

Condition Identifier: CLE\$DIVIDE_FAULT Code: CL 2069

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2070-- Exponent overflow occurred attempting {text}.

Condition Identifier: CLE\$EXPONENT_OVERFLOW Code: CL 2070

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2071-- Exponent underflow occurred attempting {text}.

Condition Identifier: CLE\$EXPONENT_UNDERFLOW Code: CL 2071

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2072-- Improper operands were used for exponentiation: {text}. The left operand may not be negative, and if the left operand is zero the right operand may not be zero or negative.

Condition Identifier: CLE\$EXPONENTIATE_FAULT Code: CL 2072

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2073-- An indefinite value resulted from attempting {text}.

Condition Identifier: CLE\$FP_INDEFINITE Code: CL 2073

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2074-- A loss of significant digits resulted from attempting {text}.

Condition Identifier: CLE\$FP_SIGNIFICANCE_LOSS Code: CL 2074

Description: Self-explanatory.

User Action: Change or conditionally evaluate the expression in order to avoid the arithmetic condition.

--ERROR CL 2075-- The "initial variable" {text} for variable {text} is not a string variable.

Condition Identifier: CLE\$INITIAL_NAME_NOT_STRING Code: CL 2075

Description: Self-explanatory.

User Action: Ensure that the type of the "initial variable" is a string.

--ERROR CL 2076-- The real number {text} cannot be converted to an integer because it is not within the range of values $-(2^{63})$ and $(2^{63})-1$.

Condition Identifier: CLE\$REAL_GREATER_THAN_INTEGER Code: CL 2076

Description: Self-explanatory.

User Action: Supply a real number within the range of possible integers.

--ERROR CL 2077-- The real number {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$REAL_NUMBER_OUT_OF_RANGE Code: CL 2077

Description: Self-explanatory.

User Action: Supply a real number in the allowed range.

--ERROR CL 2078-- The maximum value for a subrange type may not be omitted

Condition Identifier: CLE\$MAX_OF_SUBRANGE_OMITTED Code: CL 2078

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2079-- The minimum value for a subrange type must be less than or equal to the corresponding maximum value.

Condition Identifier: CLE\$MIN_OF_SUBRANGE_NOT_LE_MAX Code: CL 2079

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2080-- The default variable {text} for parameter {text} is not a string variable.

Condition Identifier: CLE\$DEFAULT_NAME_NOT_STRING Code: CL 2080

Description: Self-explanatory.

User Action: Ensure that the type of the "default variable" is string.

--ERROR CL 2081-- {text} cannot be validated to conform to {text}.

Condition Identifier: CLE\$INDETERMINATE_PARAM_VAR Code: CL 2081

Description: An element, field or substring of a variable whose type is ANY cannot be passed to a parameter with the VAR attribute.

User Action: Pass a variable of the appropriate type to the parameter.

--ERROR CL 2082-- The substring index {text} used with {text} is out of range. It must be between 1 and the maximum allowed index of {text}.

Condition Identifier: CLE\$MAX_SUBSTR_INDEX_ERROR Code: CL 2082

Description:

Self-explanatory.

User Action:

Correct the substring index.

--ERROR CL 2083-- The subscript {text} used with {text} is out of range. It must be between the maximum range {text} and {text}.

Condition Identifier: CLE\$SUBSCRIPT_OUT_OF_MAX_RANGE Code: CL 2083

Description: Self-explanatory.

User Action: Correct the subscript reference.

--ERROR CL 2084-- Only a literal quoted string is allowed for parameter {text}. No variables, function or operators may be used.

Condition Identifier: CLE\$ONLY_STRING_LITERAL_FOR_PAR Code: CL 2084

Description: Self-explanatory.

User Action: Supply a string literal (the desired characters enclosed in apostrophes).

--ERROR CL 2085-- Wrong kind of value, value out of range, or improper expression for parameter {text}: {text}.

Condition Identifier: CLE\$PARAM_EXPR_NOT_UNION_TYPE Code: CL 2085

Description: The expression given does not conform to any of the types allowed for the parameter.

User Action: Correct the expression. If you can't determine what specifically is wrong, create a variable of the type you are trying to specify and assign the expression to that variable. A message will be produced describing the error in greater detail.

--ERROR CL 2086-- The subscript {text} used with {text} is greater than the maximum number of allowed elements in the list ({text}).

Condition Identifier: CLE\$MAX_LIST_SUBSCRIPT_ERROR Code: CL 2086

Description: Self-explanatory.

User Action: Correct the subscript reference.

--ERROR CL 2087-- Values of type {text} may only be compared for equality or inequality.

Condition Identifier: CLE\$ONLY_COMPARE_FOR_EQUALITY Code: CL 2087

Description: Self-explanatory.

User Action: Change the relational operator to "=" or "<>". In the case of structured values you may compare corresponding elements or fields.

--ERROR CL 2088-- Expecting sequence number of line_identifier, found {text}.

Condition Identifier: CLE\$EXPECTING_SEQUENCE_NUMBER Code: CL 2088

Description: Self-explanatory.

User Action: Correct the line_identifier.

--ERROR CL 2089-- Range selector {text} for {text} has not been assigned a value.

Condition Identifier: CLE\$UNDEFINED_RANGE_SELECTOR Code: CL 2089

Description: Self-explanatory.

User Action: Correct the range selector.

--ERROR CL 2090-- {text} is not a keyword for parameter {text}.

Condition Identifier: CLE\$UNKNOWN_PARAMETER_KEYWORD Code: CL 2090

Description: The name is not a keyword for the parameter.

User Action: Specify a correct keyword.

--ERROR CL 2091-- {text} is not a range selector for {text}. It must be 'LOW' or 'HIGH'.

Condition Identifier: CLE\$UNKNOWN_RANGE_SELECTOR Code: CL 2091

Description: Self-explanatory.

User Action: Correct the range selector to LOW or HIGH.

--ERROR CL 2092-- Array bounds do not match.

Condition Identifier: CLE\$ARRAY_BOUNDS_DONT_MATCH Code: CL 2092

Description: The bounds of an array passed to a variable must match the bounds defined for that variable.

User Action: Supply an array with the correct bounds.

--ERROR CL 2093-- Wrong kind of value, expecting {text}, found {text} for parameter {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_PARAM_VALUE Code: CL 2093

Description: Self-explanatory.

User Action: Supply an appropriate type of value.

--ERROR CL 2094-- Specification conflict for parameter {text}: a parameter with the ADVANCED attribute may not also be a required parameter or have a default option of \$CONFIRM.

Condition Identifier: CLE\$ADVANCED_PARAMETER_CONFLICT Code: CL 2094

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2095-- The BY_NAME attribute, specified for parameter {text}, may not be used in a function declaration.

Condition Identifier: CLE\$BY_NAME_IN_FUNCTION Code: CL 2095

Description: Self-explanatory.

User Action: Remove the BY_NAME attribute.

--ERROR CL 2096-- Integer subranges do not match.

Condition Identifier: CLE\$INTEGER_RANGES_DONT_MATCH Code: CL 2096

Description: The integer subrange of an XREF/PUSH variable must match the integer subrange of the corresponding environment variable.

User Action: Supply a correct integer subrange.

--ERROR CL 2097-- Attribute conflict for parameter {text}: a parameter with the VAR attribute may not also have the DEFER attribute.

Condition Identifier: CLE\$DEFER_WITH_VAR Code: CL 2097

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2098-- Duplicate attribute {text} for parameter {text}.

Condition Identifier: CLE\$DUPLICATE_PARAMETER_ATTR Code: CL 2098

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2099-- Integer default radices do not match.

Condition Identifier: CLE\$INTEGER_RADICES_DONT_MATCH Code: CL 2099

Description: The integer radix of an XREF/PUSH variable must match the integer radix of the corresponding environment variable.

User Action: Supply the correct radix for the integer.

--ERROR CL 2100-- Expecting attribute for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAMETER_ATTR Code: CL 2100

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2101-- Expecting parameter name in {text} declaration, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAMETER_NAME Code: CL 2101

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2102-- Expecting parameter specification in {text} declaration, found {text}.

Condition Identifier: CLE\$EXPECTING_PARAMETER_SPEC Code: CL 2102

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2103-- Function parameter {text} declared with more than one name.

Condition Identifier: CLE\$FUNCTION_PARAMETER_ONE_NAME Code: CL 2103

Description: Names for function parameters may not have aliases.

User Action: Supply only one name for the function parameter.

--ERROR CL 2104-- Specification conflict for parameter {text}: a parameter with the HIDDEN attribute may not also be a required parameter or have a default option of \$CONFIRM.

Condition Identifier: CLE\$HIDDEN_PARAMETER_CONFLICT Code: CL 2104

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2105-- Type specification for function parameter {text} missing, found {text}.

Condition Identifier: CLE\$NO_TYPE_FOR_FUNCTION_PARAM Code: CL 2105

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2106-- Duplicate {text} attribute in {text} declaration.

Condition Identifier: CLE\$DUPLICATE_PROC_ATTRIBUTE Code: CL 2106

Description: Self-explanatory.

User Action: Correct the declaration.

--ERROR CL 2107-- Keywords do not match.

Condition Identifier: CLE\$KEYWORDS_DONT_MATCH Code: CL 2107

Description: Keywords specified in an XREF/PUSH variable declaration must match the keywords specified in the corresponding environment variable declaration.

User Action: Supply the correct keywords.

--ERROR CL 2108-- Expecting {text} attribute, found {text}.

Condition Identifier: CLE\$EXPECTING_PROC_ATTRIBUTE Code: CL 2108

Description: Self-explanatory.

User Action: Correct the declaration.

--ERROR CL 2109-- Expecting PROCEDURE, FUNCTION, or TYPE, found {text}.

Condition Identifier: CLE\$EXPECTING_PROC_FUNC_OR_TYPE Code: CL 2109

Description: The input to the GENERATE_PDT command must be a PROCEDURE (command) or FUNCTION declaration for a parameter description table (PDT) or a TYPE declaration for a type specification.

User Action: Correct the GENERATE_PDT input.

--ERROR CL 2110-- Expecting ";" or end of line after {text} declaration, found {text}.

Condition Identifier: CLE\$EXPECTING_PROC_HEADER_TERM Code: CL 2110

Description: Self-explanatory.

User Action: Correct the declaration.

--ERROR CL 2111-- "REST" qualifiers for the lists do not match.

Condition Identifier: CLE\$LIST_REST_DOESNT_MATCH Code: CL 2111

Description: The list "REST" qualifier of an XREF/PUSH variable must match the list "REST" qualifier of the corresponding environment variable.

User Action: Supply a list with the correct "REST" qualifier.

--ERROR CL 2112-- Function name {text} does not begin with a "\$" character.

Condition Identifier: CLE\$FUNCTION_NAME_NEEDS_\$ Code: CL 2112

Description: The first character of an SCL function must be a "\$".

User Action: Correct the function name.

--ERROR CL 2113-- Attribute conflict in {text} declaration: LOCAL may not be used with XDCL or GATE.

Condition Identifier: CLE\$PROC_SCOPE_ATTR_CONFLICT Code: CL 2113

Description: Self-explanatory.

User Action: Correct the declaration.

--ERROR CL 2114-- A string specified for a program_name may not be null or blank.

Condition Identifier: CLE\$NULL_PROGRAM_NAME Code: CL 2114

Description: Self-explanatory.

User Action: Supply a correct program name.

--ERROR CL 2115-- Program name string '{text}' is too long. It has more than 31 characters.

Condition Identifier: CLE\$PROGRAM_NAME_TOO_LONG Code: CL 2115

Description: Self-explanatory.

User Action: Supply a correct program name.

--ERROR CL 2116-- In the declaration for a RANGE type, the element type must be specified unless the declaration is for the type of a command (procedure) or function parameter.

Condition Identifier: CLE\$RANGE_ELEM_TYPE_REQUIRED Code: CL 2116

Description: Self-explanatory.

User Action: Correct the range declaration.

--ERROR CL 2117-- Expecting OF in RANGE type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_OF_FOR_RANGE Code: CL 2117

Description: Self-explanatory.

User Action: Correct the range declaration.

--ERROR CL 2118-- Expecting ")" for field list of {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_FLIST Code: CL 2118

Description: Self-explanatory.

User Action: Terminate the field list with a right parenthesis.

--ERROR CL 2119-- Wrong kind of value, value out of range, or improper expression for field {text}: {text}.

Condition Identifier: CLE\$FIELD_EXPR_NOT_UNION_TYPE Code: CL 2119

Description: The expression given does not conform to any of the types allowed for the field.

User Action: Correct the expression. If you can't determine what specifically is wrong, create a variable of the type you are trying to specify and assign the expression to that variable. A message will be produced describing the error in greater detail.

--ERROR CL 2120-- Only a literal quoted string is allowed for field {text} of {text}. No variables functions or operators may be used.

Condition Identifier: CLE\$ONLY_STRING_LITERAL_FOR_FLD Code: CL 2120

Description: Self-explanatory.

User Action: Supply a string literal (the desired characters enclosed in apostrophes).

--ERROR CL 2121-- Field {text} for {text} is required but was omitted.

Condition Identifier: CLE\$REQUIRED_FIELD_OMITTED Code: CL 2121

Description: Self-explanatory.

User Action: Supply all required fields of the record.

--ERROR CL 2122-- Too many fields for {text}.

Condition Identifier: CLE\$TOO_MANY_FIELDS Code: CL 2122

Description: The field list contains more elements than there are fields in the record.

User Action: Correct the field list.

--ERROR CL 2123-- Unexpected {text} in field list for {text}.

Condition Identifier: CLE\$UNEXPECTED_IN_FIELD_LIST Code: CL 2123

Description: Self-explanatory.

User Action: Correct the field list.

--ERROR CL 2124-- {text} is not a keyword for field {text} of {text}.

Condition Identifier: CLE\$UNKNOWN_FIELD_KEYWORD Code: CL 2124

Description: The name is not a keyword for the field.

User Action: Specify a correct keyword.

--ERROR CL 2125-- Wrong kind of value, expecting {text}, found {text} for field {text} of {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_FIELD_VALUE Code: CL 2125

Description: Self-explanatory.

User Action: Supply an appropriate type of value.

--ERROR CL 2126-- Expecting ":" after field name {text} in RECORD type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_FIELD_NAME Code: CL 2126

Description: Self-explanatory.

User Action: Correct the record declaration.

--ERROR CL 2127-- Expecting "," or ";" or end of line after specification for field {text} in RECORD type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_FIELD_SPEC Code: CL 2127

Description: Self-explanatory.

User Action: Correct the record declaration.

--ERROR CL 2128-- Expecting attribute for field {text} in RECORD type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_FIELD_ATTRIBUTE Code: CL 2128

Description: Self-explanatory.

User Action: Correct the field declaration.

--ERROR CL 2129-- Expecting \$REQUIRED or \$OPTIONAL for field {text} in RECORD type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_FIELD_REQUIREMENT Code: CL 2129

Description: Self-explanatory.

User Action: Correct the field declaration.

--ERROR CL 2130-- Expecting field name in RECORD type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_RECORD_FIELD_NAME Code: CL 2130

Description: Self-explanatory.

User Action: Correct the record declaration.

--ERROR CL 2131-- No fields in RECORD type specification.

Condition Identifier: CLE\$NO_RECORD_FIELDS Code: CL 2131

Description: A record must contain at least one field.

User Action: Correct the record declaration.

--ERROR CL 2132-- An integer for a statistic code must be between 0 and 0ffffffff(16), but {text} was given.

Condition Identifier: CLE\$STATISTIC_CODE_OUT_OF_RANGE Code: CL 2132

Description: Self-explanatory.

User Action: Correct the statistic code.

--ERROR CL 2133-- A statistic code may be specified by a name of the form "xxnnn" or "xx_nnn", where xx is the code's 2 character product identifier and nnn is an unsigned decimal number between 0 and 16777215. The following was specified: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_STATIST_NAME Code: CL 2133

Description: Self-explanatory.

User Action: Correct the statistic code name.

--ERROR CL 2134-- A statistic code may be specified by a string of the form "XX nnn", where XX is the code's 2 character product identifier and is an unsigned decimal number between 0 and 16777215. The following was specified: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_STATIST_STR Code: CL 2134

Description: Self-explanatory.

User Action: Correct the statistic code string.

--ERROR CL 2135-- An integer for a status code must be between 0 and 0ffffffff(16), but {text} was given.

Condition Identifier: CLE\$STATUS_CODE_OUT_OF_RANGE Code: CL 2135

Description: Self-explanatory.

User Action: Correct the status code.

--ERROR CL 2136-- A status code may be specified by a name of the form "xxnnn" or "xx_nnn", where xx is the code's 2 character product identifier and nnn is an unsigned decimal number between 0 and 16777215. The following was specified: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_STATUS_NAME Code: CL 2136

Description: Self-explanatory.

User Action: Correct the status code name.

--ERROR CL 2137-- A status code may be specified by a string of the form "XX nnn", where XX is the code's 2 character product identifier and nnn is an unsigned decimal number between 0 and 16777215. The following was specified: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_STATUS_STR Code: CL 2137

Description: Self-explanatory.

User Action: Correct the status code string.

--ERROR CL 2138-- The left operand of the "/" operator must be a string, a {text} was found.

Condition Identifier: CLE\$CONCAT_LEFT_OP_NOT_STR Code: CL 2138

Description: Self-explanatory.

User Action: Correct the string expression.

--ERROR CL 2139-- The right operand of the "/" operator must be a string or representable as a single string, a {text} was found.

Condition Identifier: CLE\$CONCAT_RIGHT_OP_NOT_STR Code: CL 2139

Description: Self-explanatory.

User Action: Correct the string expression.

--ERROR CL 2140-- The concatenation of strings via the "/" operator resulted in a string that exceeded the maximum string size of 65535 characters.

Condition Identifier: CLE\$CONCATENATED_STR_TOO_LONG Code: CL 2140

Description: Self-explanatory.

User Action: Shorten one or both of the strings (perhaps by using the \$TRIM function).

--ERROR CL 2141-- Only a literal quoted string is allowed. No variables, function or operators may be used.

Condition Identifier: CLE\$ONLY_STRING_LITERAL_ALLOWED Code: CL 2141

Description: Self-explanatory.

User Action: Supply a string literal (the desired characters enclosed in apostrophes).

--ERROR CL 2142-- String '{text}' should not have more than {text} characters in it but contains {text}.

Condition Identifier: CLE\$STRING_VALUE_TOO_LONG Code: CL 2142

Description: Self-explanatory.

User Action: Correct the string value.

--ERROR CL 2143-- String '{text}' should have at least {text} characters in it but contains only {text}.

Condition Identifier: CLE\$STRING_VALUE_TOO_SHORT Code: CL 2143

Description: Self-explanatory.

User Action: Correct the string value.

--ERROR CL 2144-- List size qualifiers do not match.

Condition Identifier: CLE\$LIST_SIZES_DONT_MATCH Code: CL 2144

Description: The list size qualifier of an XREF/PUSH variable must match the list size qualifier of the corresponding environment variable.

User Action: Supply the correct list size qualifier.

--ERROR CL 2145-- The minimum size for a string may not be greater than the corresponding maximum size.

Condition Identifier: CLE\$MIN_STRING_SIZE_GT_MAX Code: CL 2145

Description: Self-explanatory.

User Action: Correct the string declaration.

--ERROR CL 2146-- The string size {text} is out of range. It must be between {text} and {text}.

Condition Identifier: CLE\$STRING_SIZE_OUT_OF_RANGE Code: CL 2146

Description: Self-explanatory.

User Action: Correct the string declaration.

--ERROR CL 2147-- Expecting ")" of subscript qualifier used with {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_SUBSCR Code: CL 2147

Description: Self-explanatory.

User Action: Terminate the subscript qualifier with a right parenthesis.

--ERROR CL 2148-- The subscript {text} used with {text} is greater than the current number of elements in the list ({text}).

Condition Identifier: CLE\$LIST_SUBSCRIPT_TOO_LARGE Code: CL 2148

Description: Self-explanatory.

User Action: Correct the subscript value.

--ERROR CL 2149-- The subscript {text} used with {text} is too small to be used with a list. A list subscript must be greater than or equal to 1.

Condition Identifier: CLE\$LIST_SUBSCRIPT_TOO_SMALL Code: CL 2149

Description: Self-explanatory.

User Action: Correct the subscript value.

--ERROR CL 2150-- Name sizes do not match.

Condition Identifier: CLE\$NAME_SIZES_DONT_MATCH Code: CL 2150

Description: The name size of an XREF/PUSH variable must match the name size of the corresponding environment variable.

User Action: Supply the correct name size.

--ERROR CL 2151-- Element {text} within {text} has not been assigned a value.

Condition Identifier: CLE\$UNDEFINED_SUBSCR_ELEMENT Code: CL 2151

Description: Self-explanatory.

User Action: Correct the subscript reference.

--ERROR CL 2152-- Expecting ")" of substring qualifier used with {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_RPAREN_OF_SUBSTR Code: CL 2152

Description: Self-explanatory.

User Action: Terminate the substring qualifier with a right parenthesis.

--ERROR CL 2153-- The substring index {text} used with {text} is out of range. It must be between 1 and {text}.

Condition Identifier: CLE\$SUBSTR_INDEX_OUT_OF_RANGE Code: CL 2153

Description: Self-explanatory.

User Action: Correct the substring index.

--ERROR CL 2154-- The substring size {text} used with {text} is out of range. It must be between 0 and {text}.

Condition Identifier: CLE\$SUBSTR_SIZE_OUT_OF_RANGE Code: CL 2154

Description: Self-explanatory.

User Action: Correct the substring size.

--ERROR CL 2155-- Expecting DATE_TIME after "-" in TIME_INCREMENT expression, found {text}.

Condition Identifier: CLE\$EXPECTING_DATE_TIME_OPERAND Code: CL 2155

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2156-- Expecting "-" after DATE_TIME in TIME_INCREMENT expression, found {text}.

Condition Identifier: CLE\$EXPECTING_DATE_TIME_SUBTRCT Code: CL 2156

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2157-- Expecting TIME_INCREMENT expression, found {text}.

Condition Identifier: CLE\$EXPECTING_TIME_INCR_EXPR Code: CL 2157

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2158-- Expecting TIME_INCREMENT variable or function after {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_TIME_INCR_OPERAND Code: CL 2158

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2159-- Expecting TIME_ZONE after "-" in TIME_INCREMENT expression, found {text}.

Condition Identifier: CLE\$EXPECTING_TIME_ZONE_OPERAND Code: CL 2159

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2160-- Expecting "-" after TIME_ZONE in TIME_INCREMENT expression, found {text}.

Condition Identifier: CLE\$EXPECTING_TIME_ZONE_SUBTRCT Code: CL 2160

Description: Self-explanatory.

User Action: Correct the time_increment expression.

--ERROR CL 2161-- Unrecognizable TIME_INCREMENT expression: {text}.

Condition Identifier: CLE\$UNRECOGNIZABLE_TIME_INCR Code: CL 2161

Description: Self-explanatory.

User Action: Correct the time_increment value.

--ERROR CL 2162-- Expecting TIME_ZONE expression, found {text}.

Condition Identifier: CLE\$EXPECTING_TIME_ZONE_EXPR Code: CL 2162

Description: Self-explanatory.

User Action: Correct the time_zone expression.

--ERROR CL 2163-- Wrong kind of range type, expecting {text}, found {text}.

Condition Identifier: CLE\$RANGE_TYPES_DONT_MATCH Code: CL 2163

Description: The type qualifier of a range passed to a variable must match the type qualifier defined for that range variable.

User Action: Supply the correct range type.

--ERROR CL 2164-- Real subranges do not match.

Condition Identifier: CLE\$REAL_SUBRANGES_DONT_MATCH Code: CL 2164

Description: The real subrange of an XREF/PUSH variable must match the real subrange of the corresponding environment variable.

User Action: Supply a real with the correct subrange.

--ERROR CL 2165-- Expecting comma, semicolon or end of line after TYPE definition, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_TYPE_DEF Code: CL 2165

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2166-- Expecting : or = after TYPE name, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_TYPE_NAME Code: CL 2166

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2167-- Expecting TYPE name, found {text}.

Condition Identifier: CLE\$EXPECTING_TYPE_NAME Code: CL 2167

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2168-- Expecting TYPEND, found {text}.

Condition Identifier: CLE\$EXPECTING_TYPEND Code: CL 2168

Description: Only one TYPE declaration may be specified for the input to GENERATE_PDT at a time.

User Action: Correct the type declaration.

--ERROR CL 2169-- Expecting "," or ";" or end of line after member type specification in ANY specification, found {text}.

Condition Identifier: CLE\$EXPECTING_AFTER_MEMBER_SPEC Code: CL 2169

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2170-- Expecting OF in ANY type specification, found {text}.

Condition Identifier: CLE\$EXPECTING_OF_FOR_ANY Code: CL 2170

Description: Self-explanatory.

User Action: Correct the type declaration.

--ERROR CL 2171-- Numeric member types of a union type specified different radices. If multiple integer types are specified they must have identical radices. If integer and real types are specified, the radix for the integer types must be 10.

Condition Identifier: CLE\$INCONSISTENT_RADIX_IN_UNION Code: CL 2171

Description: Self-explanatory.

User Action: Correct the ANY type declaration.

--ERROR CL 2172-- No member types in ANY type specification.

Condition Identifier: CLE\$NO_UNION_MEMBERS Code: CL 2172

Description: An "ANY OF" type must contain at least one member type.

User Action: Correct the "ANY OF" type declaration.

--ERROR CL 2173-- Field names for records do not match.

Condition Identifier: CLE\$FIELD_NAMES_DONT_MATCH Code: CL 2173

Description: The field names of a record passed to a variable must match the field names defined for that variable.

User Action: Supply a record with the correct field names.

--ERROR CL 2174-- A reference was made to variable {text}, but no value has ever been assigned to it.

Condition Identifier: CLE\$VARIABLE_NEVER_GIVEN_VALUE Code: CL 2174

Description: Self-explanatory.

User Action: Assign a value to the variable.

--ERROR CL 2175-- Field requirements for records do not match.

Condition Identifier: CLE\$FIELD_REQUIREMENTS_MISMATCH Code: CL 2175

Description: The field requirements of a record passed to a variable must match the field requirements defined for that variable.

User Action: Supply a record with the correct field requirements.

--ERROR CL 2176-- The VAR attribute, specified for parameter {text}, may not be used in a function declaration.

Condition Identifier: CLE\$VAR_PARAM_IN_FUNCTION Code: CL 2176

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2177-- Attribute conflict for parameter {text}: a parameter with the VAR attribute may not also have the SECURE attribute.

Condition Identifier: CLE\$SECURE_WITH_VAR Code: CL 2177

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2178-- The SECURE attribute, specified for parameter {text}, may not be used in a function declaration.

Condition Identifier: CLE\$SECURE_PARAM_IN_FUNCTION Code: CL 2178

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2179-- Expecting variable for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_VAR_FOR_PARAM Code: CL 2179

Description: Self-explanatory.

User Action: Supply a variable for the parameter.

--ERROR CL 2180-- Wrong kind of variable, expecting {text}, found {text} for parameter {text}.

Condition Identifier: CLE\$WRONG_KIND_OF_PARAM_VAR Code: CL 2180

Description: Self-explanatory.

User Action: Supply an appropriate type of variable.

--ERROR CL 2181-- Wrong kind of record field type, expecting {text}, found {text} for field {text}.

Condition Identifier: CLE\$FIELD_TYPES_DONT_MATCH Code: CL 2181

Description: The field types of a record passed to a variable must match the field types defined for that variable.

User Action: Supply a record with the correct field type specifications.

--ERROR CL 2182-- Number of fields for records do not match.

Condition Identifier: CLE\$NUMBER_OF_FIELDS_DONT_MATCH Code: CL 2182

Description: The number of fields of a record passed to a variable must match the number of fields defined for that variable.

User Action: Supply a record with the correct number of fields.

--ERROR CL 2183-- "LITERAL" qualifier for strings do not match.

Condition Identifier: CLE\$STRING_LITERALS_DONT_MATCH Code: CL 2183

Description: The "LITERAL" qualifier of a string XREF/PUSH variable must match the "LITERAL" qualifier of the corresponding environment variable.

User Action: Supply a string with the correct "LITERAL" qualifier.

--ERROR CL 2184-- String sizes do not match.

Condition Identifier: CLE\$STRING_SIZES_DONT_MATCH Code: CL 2184

Description: The string size specification of an XREF/PUSH variable must match the string size specification of the corresponding environment variable.

User Action: Supply a string with the correct size.

--ERROR CL 2185-- Application type names do not match.

Condition Identifier: CLE\$APPLICATION_NAME_MISMATCH Code: CL 2185

Description: Application type name passed to a variable must match the application type name defined for that variable.

User Action: Supply a correct application type name.

--ERROR CL 2186-- "BALANCE_BRACKETS" attributes do not match.

Condition Identifier: CLE\$BALANCE_BRACKETS_DONT_MATCH Code: CL 2186

Description: The "BALANCE_BRACKETS" attributes of an application passed to a variable must match the "BALANCE_BRACKETS" attributes defined for that variable.

User Action: Supply an application with the correct "BALANCE_BRACKETS" attribute.

--ERROR CL 2187-- A deferred value cannot be assigned to a component of variable {text}.

Condition Identifier: CLE\$IMPROPER_USE_OF_DEFER_VALUE Code: CL 2187

Description: Self-explanatory.

User Action: Assign a value that is not a deferred value.

--ERROR CL 2188-- Cannot assign a value to a component of variable {text} because it has the deferred attribute.

Condition Identifier: CLE\$IMPROPER_USE_OF_DEFER_VAR Code: CL 2188

Description:

A variable defined with the DEFER attribute cannot have its value changed by a component.

User Action:

Assign a value to the entire variable instead of a component.

--ERROR CL 2189-- Date_Time tenses do not match.

Condition Identifier: CLE\$DATE_TIME_TENSES_DONT_MATCH Code: CL 2189

Description: The date_time tense of an XREF/PUSH variable must match the date_time tense of the corresponding environment variable.

User Action: Supply a date_time with the correct tense.

--ERROR CL 2190-- Date_Time type identifiers do not match.

Condition Identifier: CLE\$DATE_TIME_TYPES_DONT_MATCH Code: CL 2190

Description: The date_time type specification of an XREF/PUSH variable must match the date_time type specification of the corresponding environment variable.

User Action: Supply a date_time with the correct type specification.

--ERROR CL 2191-- Undefined array element type.

Condition Identifier: CLE\$UNKNOWN_ARRAY_ELEMENT_TYPE Code: CL 2191

Description: The array element type specification is undefined.

User Action: Supply an array element type specification.

--ERROR CL 2192-- Undefined list element type.

Condition Identifier: CLE\$UNKNOWN_LIST_ELEMENT_TYPE Code: CL 2192

Description: The list element type specification is undefined.

User Action: Supply a list element type specification.

--ERROR CL 2193-- Type is not defined.

Condition Identifier: CLE\$UNDEFINED_TYPE Code: CL 2193

Description: The type specification is undefined.

User Action: Supply a type specification.

--ERROR CL 2194-- Value is not defined.

Condition Identifier: CLE\$UNDEFINED_VALUE Code: CL 2194

Description: The value is undefined.

User Action: Supply a value.

--ERROR CL 2195-- Wrong kind of value or value out of range for a union (ANY) type.

Condition Identifier: CLE\$VALUE_NOT_UNION_TYPE Code: CL 2195

Description: The value passed to a union (ANY) variable does not conform to the member types defined for that variable.

User Action: Supply the correct value for the union.

--ERROR CL 2196-- Undefined range element type.

Condition Identifier: CLE\$UNKNOWN_RANGE_ELEMENT_TYPE Code: CL 2196

Description: The range element type is undefined.

User Action: Supply a range element type.

--ERROR CL 2198-- Member types of union do not match for parameter {text}.

Condition Identifier: CLE\$P_ANY_MEMBER_TYPE_MISMATCH Code: CL 2198

Description: No reference.

User Action:

--ERROR CL 2199-- Application type names do not match for parameter {text}.

Condition Identifier: CLE\$P_APPLICATION_NAME_MISMATCH Code: CL 2199

Description: The application type name passed to a parameter must match the application type name defined for that parameter.

User Action: Supply a correct application type name.

--ERROR CL 2200-- Array bounds do not match for parameter {text}.

Condition Identifier: CLE\$P_ARRAY_BOUNDS_DONT_MATCH Code: CL 2200

Description: The bounds of an array passed to a parameter must match the bounds defined for that parameter.

User Action: Supply an array with the correct bounds.

--ERROR CL 2201-- "BALANCE_BRACKETS" attributes do not match for parameter {text}.

Condition Identifier: CLE\$P_BALANCE_BRACKETS_MISMATCH Code: CL 2201

Description: The "BALANCE_BRACKETS" attribute of an application passed to a parameter must match the "BALANCE_BRACKETS" attribute defined for that parameter.

User Action: Supply an application with the correct "BALANCE_BRACKETS" attribute.

--ERROR CL 2202-- Date_Time tenses do not match for parameter {text}.

Condition Identifier: CLE\$P_DATE_TIME_TENSES_MISMATCH Code: CL 2202

Description: The tense of a date_time passed to a (VAR) parameter must match the tense defined for that parameter.

User Action: Supply a date_time with the correct tense.

--ERROR CL 2203-- Date_Time type identifiers do not match for parameter {text}.

Condition Identifier: CLE\$P_DATE_TIME_TYPES_MISMATCH Code: CL 2203

Description: The type specification of a date_time passed to a (VAR) parameter must match the type specification defined for that parameter.

User Action: Supply a date_time with the correct type specification.

--ERROR CL 2204-- Field names for records do not match for parameter {text}.

Condition Identifier: CLE\$P_FIELD_NAMES_DONT_MATCH Code: CL 2204

Description: The field names of a record passed to a parameter must match the field names defined for that parameter.

User Action: Supply a record with the correct field names.

--ERROR CL 2205-- Field requirements for records do not match for parameter {text}.

Condition Identifier: CLE\$P_FIELD_REQUIREMNT_MISMATCH Code: CL 2205

Description: The field requirements of a record passed to a parameter must match the field requirements defined for that parameter.

User Action: Supply a record with the correct field requirements.

--ERROR CL 2206-- Wrong kind of record field type, expecting {text}, found {text} for field {text} for parameter {text}.

Condition Identifier: CLE\$P_FIELD_TYPES_DONT_MATCH Code: CL 2206

Description: The field type specification of a record passed to a parameter must match the field type specification defined for that parameter.

User Action: Supply a record with the correct field type specifications.

--ERROR CL 2207-- Integer default radices do not match for parameter {text}.

Condition Identifier: CLE\$P_INTEGER_RADICES_MISMATCH Code: CL 2207

Description: The radix of an integer passed to a (VAR) parameter must match the radix defined for that parameter.

User Action: Supply the correct radix for the integer.

--ERROR CL 2208-- Integer subranges do not match for parameter {text}.

Condition Identifier: CLE\$P_INTEGER_RANGES_DONT_MATCH Code: CL 2208

Description: The subrange of an integer passed to a (VAR) parameter must match the subrange defined for that parameter.

User Action: Supply a correct integer subrange.

--ERROR CL 2209-- Keywords do not match for parameter {text}.

Condition Identifier: CLE\$P_KEYWORDS_DONT_MATCH Code: CL 2209

Description: A keyword passed to a (VAR) parameter must match the keywords defined for that parameter.

User Action: Supply the correct keywords.

--ERROR CL 2210-- "REST" qualifiers for the lists do not match for parameter {text}.

Condition Identifier: CLE\$P_LIST_REST_DOESNT_MATCH Code: CL 2210

Description: The "REST" qualifier of a list passed to a (VAR) parameter must match the "REST" qualifier defined for that parameter.

User Action: Supply the correct list size qualifier.

--ERROR CL 2211-- List size qualifiers do not match for parameter {text}.

Condition Identifier: CLE\$P_LIST_SIZES_DONT_MATCH Code: CL 2211

Description: The size qualifier of a list passed to a (VAR) parameter must match the size qualifier defined for that parameter.

User Action: Supply the correct list size qualifier.

--ERROR CL 2212-- Name sizes do not match for parameter {text}.

Condition Identifier: CLE\$P_NAME_SIZES_DONT_MATCH Code: CL 2212

Description: The size specification of a name passed to a (VAR) parameter must match the size specification defined for that parameter.

User Action: Supply the correct range type.

--ERROR CL 2213-- Number of fields for records do not match for parameter {text}.

Condition Identifier: CLE\$P_NUMBER_OF_FIELDS_MISMATCH Code: CL 2213

Description: The number of fields of a record passed to a parameter must match the number of fields defined for that parameter.

User Action: Supply a record with the correct number of fields.

--ERROR CL 2214-- Wrong kind of range type, expecting {text}, found {text} for parameter {text}.

Condition Identifier: CLE\$P_RANGE_TYPES_DONT_MATCH Code: CL 2214

Description: The type specification of a range passed to a parameter must match the type specification defined for that parameter.

User Action: Supply the correct range type.

--ERROR CL 2215-- Real subranges do not match for parameter {text}.

Condition Identifier: CLE\$P_REAL_SUBRANGES_DONT_MATCH Code: CL 2215

Description: The subrange of a real passed to a (VAR) parameter must match the subrange defined for that parameter.

User Action: Supply a real with the correct subrange.

--ERROR CL 2216-- "LITERAL" qualifier for strings do not match for parameter {text}.

Condition Identifier: CLE\$P_STRING_LITERAL_MISMATCH Code: CL 2216

Description: The "LITERAL" qualifier of a string passed to a (VAR) parameter must match the "LITERAL" qualifier defined for that parameter.

User Action: Supply a string with the correct "LITERAL" qualifier.

--ERROR CL 2217-- String sizes do not match for parameter {text}.

Condition Identifier: CLE\$P_STRING_SIZES_DONT_MATCH Code: CL 2217

Description: The size specification of a string passed to a (VAR) parameter must match the size specification defined for that parameter.

User Action: Supply a string with the correct size.

--ERROR CL 2218-- Type is not defined for parameter {text}.

Condition Identifier: CLE\$P_UNDEFINED_TYPE Code: CL 2218

Description: The type specification passed to the parameter is undefined.

User Action: Supply a type specification.

--ERROR CL 2219-- Value is not defined for parameter {text}.

Condition Identifier: CLE\$P_UNDEFINED_VALUE Code: CL 2219

Description: The value passed to the parameter is undefined.

User Action: Supply a value.

--ERROR CL 2220-- Undefined array element type for parameter {text}.

Condition Identifier: CLE\$P_UNKNOWN_ARRAY_ELEM_TYPE Code: CL 2220

Description: The element type specification of the array passed to the parameter is undefined.

User Action: Supply an array element type specification.

--ERROR CL 2221-- Undefined list element type for parameter {text}.

Condition Identifier: CLE\$P_UNKNOWN_LIST_ELEMENT_TYPE Code: CL 2221

Description: The element type specification of the list passed to the parameter is undefined.

User Action: Supply a list element type specification.

--ERROR CL 2222-- Undefined range element type for parameter {text}.

Condition Identifier: CLE\$P_UNKNOWN_RANGE_ELEM_TYPE Code: CL 2222

Description: The element type specification of the range passed to the parameter is undefined.

User Action: Supply a range element type specification.

--ERROR CL 2223-- Duplicate field name {text} in RECORD type specification.

Condition Identifier: CLE\$DUPLICATE_FIELD_NAME Code: CL 2223

Description: Self-explanatory.

User Action: Correct the record declaration.

--ERROR CL 2224-- Required parameter {text} was specified with omitted procedure parameter {text}.

Condition Identifier: CLE\$REQUIRED_PARAMETER_UNSPEC Code: CL 2224

Description: Self-explanatory.

User Action: Ensure that an optional procedure parameter was specified before attempting to pass it to a required parameter of another command or function.

--ERROR CL 2225-- Required field {text} for {text} was specified with omitted procedure parameter {text}.

Condition Identifier: CLE\$REQUIRED_FIELD_UNSPECIFIED Code: CL 2225

Description: Self-explanatory.

User Action: Ensure that an optional procedure parameter was specified before attempting to assign it to a required field of a record.

--ERROR CL 2226-- No elements given for array, but at least one element must be given.

Condition Identifier: CLE\$UNEXPECTED_EMPTY_ARRAY Code: CL 2226

Description: Self-explanatory.

User Action: Supply the correct number of elements for the array.

--ERROR CL 2227-- Expecting semicolon or end of line after specification for parameter {text}, found {text}.

Condition Identifier: CLE\$EXPECTING_END_OF_PARAM_SPEC Code: CL 2227

Description: Self-explanatory.

User Action: Correct the parameter declaration.

--ERROR CL 2228-- Wrong kind of variable for a union (ANY) type for parameter {text}.

Condition Identifier: CLE\$VARIABLE_NOT_UNION_TYPE Code: CL 2228

Description: The variable passed to a union (ANY) parameter does not conform to the member types defined for that parameter.

User Action: Supply a variable which conforms to union (ANY) type of parameter.

--ERROR CL 2229-- Wrong kind of value or value out of range for a union (ANY) type for parameter {text}.

Condition Identifier: CLE\$P_VALUE_NOT_UNION_TYPE Code: CL 2229

Description: The value passed to a union (ANY) parameter does not conform to the member types defined for that parameter.

User Action: Supply a value which conforms to the union (ANY) type of the parameter.

--ERROR CL 2231-- A reference was made to parameter {text}, but no value has ever been assigned to it.

Condition Identifier: CLE\$PARAMETER_NEVER_GIVEN_VALUE Code: CL 2231

Description: A value must be assigned to a parameter before that parameter can be referenced.

User Action: Pass a value into the parameter before referencing it.

--ERROR CL 5025-- {text} compare errors.

Condition Identifier: CLE\$COMPARE_ERRORS_DETECTED Code: CL 5025

Description: The comparison found the specified number of differences in the contents of the compared files.

User Action: None.

--ERROR CL 5030-- File {file} shorter than file {file}.

Condition Identifier: CLE\$COMPARED_FILES_UNEQUAL_SIZE Code: CL 5030

Description: The two files compared have different lengths.

User Action: None.

--INFORMATIVE CL 7000-- Welcome to the NOS/VE Software System. Copyright Control Data 1983, 1988. {text} SN{text}. {text} {text}. {text}.

Condition Identifier: CLE\$WELCOME_BANNER Code: CL 7000

Description: This message is issued upon successful logon to NOS/VE.

User Action: None.

--INFORMATIVE CL 7001-- You have the following detached jobs: {text}

Condition Identifier: CLE\$DETACHED_JOBS Code: CL 7001

Description: This message is issued at the beginning of an interactive job if the user has any detached jobs (jobs resulting from a DETACH_JOB command, line disconnect, etc). The system supplied names of all such jobs are included in the message.

User Action: The user may reconnect to any one of the detached jobs by using the ATTACH_JOB command

--INFORMATIVE CL 7002-- Command terminated.

Condition Identifier: CLE\$COMMAND_TERMINATED Code: CL 7002

Description: This message is issued upon successful termination of a command via a terminate break, or via the TERMINATE_COMMAND command.

User Action: None.

--INFORMATIVE CL 7003-- End of information from terminal file ignored.

Condition Identifier: CLE\$INTERACTIVE_EOI_IGNORED Code: CL 7003

Description: In response to a command prompt, the user entered the string defined by the END_OF_INFORMATION terminal attribute. The entry has been ignored.

User Action: Enter a command. If the intent was to end the interactive job, enter the LOGOUT command.

--INFORMATIVE CL 7004-- End of partition from terminal file ignored.

Condition Identifier: CLE\$INTERACTIVE_EOP_IGNORED Code: CL 7004

Description: In response to a command prompt, the user entered the string defined by the End Of Information terminal attribute. The entry has been ignored.

User Action: Enter a command. If the intent was to end the interactive job, enter the LOGOUT command.

--INFORMATIVE CL 7014-- Command line cancelled.

Condition Identifier: CLE\$COMMAND_LINE_CANCELLED Code: CL 7014

Description: The processing of a command line entered in response to a parameter prompt was cancelled by a terminate break.

User Action: No action required.

--INFORMATIVE CL 7015-- Command {text} cancelled.

Condition Identifier: CLE\$COMMAND_CANCELLED Code: CL 7015

Description: The processing of the specified command was cancelled by the entry of terminate break in response to a parameter prompt.

User Action: No action required.

--INFORMATIVE CL 7018-- Your login password will expire {text} {text}.

Condition Identifier: CLE\$PASSWORD_EXPIRATION_WARNING Code: CL 7018

Description: You will not be able to log in to the system if you do not change your password before it expires.

User Action: Use the CHANGE_LOGIN_PASSWORD command to change your password.

--ERROR CL 8003-- Bad help module.

Condition Identifier: CLE\$BAD_HELP_MODULE Code: CL 8003

Description: The help module specified on the request could not be searched.

User Action: Check that the help module is valid.

--ERROR CL 8005-- Condition code is too large for {text}.

Condition Identifier: CLE\$COND_CODE_TOO_LARGE Code: CL 8005

Description: A condition number may not be greater than 999999.

User Action: Correct the status message code definition.

--ERROR CL 8010-- {text} defined more than once. First definition is used.

Condition Identifier: CLE\$CONSTANT_ALREADY_DEFINED Code: CL 8010

Description: GENERATE_MESSAGE_TEMPLATES encountered a duplicate definition of a condition name or identifier used to define condition names. The duplicate definition is ignored.

User Action: Change the input to eliminate duplicates.

--ERROR CL 8015-- {text} is not a defined constant when defining {text}.

Condition Identifier: CLE\$CONSTANT_NOT_DEFINED Code: CL 8015

Description: An identifier for an undefined constant was referenced in the definition of another constant.

User Action: Order the definitions so that constants are defined before they are referenced.

--ERROR CL 8020-- Constant save area overflow when saving {text}.

Condition Identifier: CLE\$CONSTANT_STACK_OVERFLOW Code: CL 8020

Description: The combined number of condition names and identifiers used to define condition names has exceeded the storage area used to save their constant definitions.

User Action: Reduce the number of condition names or identifiers used to define the condition names.

--ERROR CL 8025-- Duplicated condition code {text} found in module {text}.

Condition Identifier: CLE\$DUPLICATE_CONDITION_CODES Code: CL 8025

Description: A message module was specified which contained duplicate definitions for the specified status condition code. The module was created with the duplicates ignored.

User Action: Change the message definitions to eliminate the duplicates.

--ERROR CL 8026-- Key {text} has already been assigned to a menu item.

Condition Identifier: CLE\$DUPLICATE_KEYS Code: CL 8026

Description: Self-explanatory.

User Action: Change the key used for the menu item to eliminate duplication.

--ERROR CL 8027-- Duplicate menu class : {text}.

Condition Identifier: CLE\$DUPLICATE_MENU_CLASS Code: CL 8027

Description: Self-explanatory.

User Action: Change the name to eliminate the duplication.

--ERROR CL 8030-- {text} entered more than once for {text} in module {text}.

Condition Identifier: CLE\$DUPLICATE_NAMES Code: CL 8030

Description: A duplicate name has been encountered for the same kind of message or menu.

User Action: Change the name to eliminate the duplication.

--ERROR CL 8035-- Attempt to enter {text} but one is already defined for module {text}.

Condition Identifier: CLE\$DUPLICATE_HELP_MESSAGES Code: CL 8035

Description: A message or help module was specified that contained duplicate definitions for the specified messages. The module was erected with the duplicates ignored.

User Action: Change the message definition to eliminate the duplicates.

--ERROR CL 8037-- Shifted key {text} has already been assigned to a menu item.

Condition Identifier: CLE\$DUPLICATE_SHIFTED_KEYS Code: CL 8037

Description: Self-explanatory.

User Action: Change the menu item key to eliminate the duplication.

--WARNING CL 8040-- Errors encountered in creation of message module {text}.

Condition Identifier: CLE\$ERRORS_IN_MODULE Code: CL 8040

Description: Errors in the definitions for messages in a message module were encountered.

User Action: Correct errors in the message definition.

--ERROR CL 8045-- Expecting "=", found {text} when defining {text}.

Condition Identifier: CLE\$EXPECTING_EQUAL_SIGN Code: CL 8045

Description: An incorrect symbol was found after the name of a constant within its definition.

User Action: Correct the constant definition.

--ERROR CL 8050-- Expecting integer value, found {text} when defining {text}.

Condition Identifier: CLE\$EXPECTING_INTEGER_VALUE Code: CL 8050

Description: Constants for status conditions must be defined with integer values.

User Action: Correct the constant definitions.

--ERROR CL 8055-- Expecting MODULE name value, found {text}.

Condition Identifier: CLE\$EXPECTING_MODULE_NAME Code: CL 8055

Description: No module name follow the word MODULE.

User Action: Correct the MODULE name specification.

--ERROR CL 8060-- Expecting name for constant or code definition, found {text}.

Condition Identifier: CLE\$EXPECTING_NAME_VALUE Code: CL 8060

Description: A status condition constant definition must begin with a name.

User Action: Correct the constant definition.

--WARNING CL 8065-- Internal generator error after defining {text}.

Condition Identifier: CLE\$INTERNAL_GENERATOR_ERROR Code: CL 8065

Description: The message template generator encountered a problem that was internal to the generator itself.

User Action: Write a PSR and include the appropriate supporting documentation.

--WARNING CL 8070-- {text} is not defined as an integer constant.

Condition Identifier: CLE\$INVALID_INTEGER_CONSTANT Code: CL 8070

Description: A constant for a status condition code must be defined as an integer.

User Action: Correct the definition.

--WARNING CL 8072-- {text} is not defined as an integer or string constant.

Condition Identifier: CLE\$INVALID_INT_OR_STRING_CONST Code: CL 8072

Description: Only integer or string constants may be defined within the input to GENERATE_MESSAGE_TEMPLATES.

User Action: Correct the definition.

--ERROR CL 8073-- Maximum number of menu classes exceeded - the maximum is 16.

Condition Identifier: CLE\$MAX_MENU_CLASSES_EXCEEDED Code: CL 8073

Description: Self-explanatory.

User Action: Reduce the number of menu classes.

--ERROR CL 8074-- Maximum number of menu items per class exceeded - the maximum is 20.

Condition Identifier: CLE\$MAX_MENU_ITEMS_EXCEEDED Code: CL 8074

Description: Self-explanatory.

User Action: Reduce the number of menu items.

--ERROR CL 8075-- Menu class was not previously defined: {text}.

Condition Identifier: CLE\$MENU_CLASS_NOT_DEFINED Code: CL 8075

Description: A menu class must be defined before associating it with a menu item.

User Action: Define the menu class by using the create_menu_class command.

--FATAL CL 8077-- Message module {text} is too large.

Condition Identifier: CLE\$MODULE_TOO_LARGE Code: CL 8077

Description: The message module is too large.

User Action: Split up the message definitions to create two or more message modules.

--ERROR CL 8078-- No menu class has been defined.

Condition Identifier: CLE\$NO_MENU_CLASS_DEFINED Code: CL 8078

Description: Self-explanatory.

User Action: Define a menu class via the create_menu_class command.

--ERROR CL 8080-- No message text found for {text}.

Condition Identifier: CLE\$NO_MESSAGE_TEXT Code: CL 8080

Description: No text for the message corresponding to the stated condition name was found by GENERATE_MESSAGE_TEMPLATES.

User Action: Correct the condition's defining text.

--ERROR CL 8085-- No messages defined for module {text}.

Condition Identifier: CLE\$NO_MODULE_CREATED Code: CL 8085

Description: No messages were created with the stated message module. A message module must contain at least one message.

User Action: Define a message within the module.

--ERROR CL 8090-- No severity level found for {text}. "E" is assumed.

Condition Identifier: CLE\$NO_SEVERITY_LEVEL Code: CL 8090

Description: No severity for the stated condition name was found by GENERATE_MESSAGE_TEMPLATES.

User Action: Correct the condition's defining text.

--ERROR CL 8092-- Null name is not valid for {text}.

Condition Identifier: CLE\$NULL_NOT_ALLOWED Code: CL 8092

Description: Self-explanatory.

User Action: Do not use the null name.

--WARNING CL 8095-- Template defined for {text} in module {text} exceeds maximum allowable length of {text} characters. Template was truncated.

Condition Identifier: CLE\$TEMPLATE_TOO_LONG Code: CL 8095

Description: The template defined in the module exceeds maximum length. The template was created but truncated.

User Action: Decrease the message size.

--ERROR CL 8097-- Too few classes defined for application menu {text}.

Condition Identifier: CLE\$TOO_FEW_CLASSES Code: CL 8097

Description: Self-explanatory.

User Action: Define at least one menu class via the create_menu_class command.

--ERROR CL 8099-- Too few items defined for application menu {text}.

Condition Identifier: CLE\$TOO_FEW_ITEMS Code: CL 8099

Description: Self-explanatory.

User Action: Define at least one menu item via the create_menu_item command.

--ERROR CL 8100-- Number of entries exceeds maximum of {text} for message module {text}.

Condition Identifier: CLE\$TOO_MANY_ENTRIES_FOR_MODULE Code: CL 8100

Description: The number of entries exceeds the maximum allowed for a message module.

User Action: Split up the message definitions to create two or more message modules.

--ERROR CL 8105-- {text} is not defined as a recognizable exception condition code base.

Condition Identifier: CLE\$UNRECOGNIZABLE_ECC_BASE Code: CL 8105

Description: Self-explanatory.

User Action: Correct the definition of the condition code base.

--ERROR CL 9005-- Expecting {text} entry, found {text} entry.

Condition Identifier: CLE\$UNEXPECTED_ENTRY Code: CL 9005

Description: Within GENERATE_COMMAND_TABLE, definitions for commands and and functions may not appear in the same table.

User Action: Separate the definitions for commands and functions into their own respective tables.

--ERROR CL 9010-- Improper module name - {text}.

Condition Identifier: CLE\$IMPROPER_MODULE_NAME Code: CL 9010

Description: The module name must conform to specifications for CYBIL identifiers.

User Action: Correct the module name.

--WARNING CL 9015-- Name "{text}" duplicated in command or function table.

Condition Identifier: CLE\$DUPLICATE_CMND_OR_FCN_NAME Code: CL 9015

Description: There is a duplicate command or function name.

User Action: Check to see that names chosen for all names, aliases, and abbreviations are unique within a command or function table.

--ERROR CL 9510-- {text} can not be translated.

Condition Identifier: CLE\$CANNOT_BE_TRANSLATED Code: CL 9510

Description: The item in the message cannot be translated from the "old style" usage to the new, since the capability it represents is no longer supported.

User Action: Consult the "System Usage" and "Commands and Functions" manuals for an alternative capability.

--ERROR CL 9520-- Utility name {text} already specified.

Condition Identifier: CLE\$DUPLICATE_UTILITY_NAME Code: CL 9520

Description: Within a utility definition file for the SCL formatter, a definition for the utility name defined was already found.

User Action: Change the file to eliminate the duplication.

--ERROR CL 9540-- {text} Error messages. {text} Warning messages.

Condition Identifier: CLE\$ERRORS_AND_WARNINGS Code: CL 9540

Description: A command found the specified number of errors and warnings in its input.

User Action: Correct the input for the command.

--ERROR CL 9560-- {text} Error messages.

Condition Identifier: CLE\$ERRORS_ENCOUNTERED Code: CL 9560

Description: A command found the specified number of errors in its input.

User Action: Correct the input for the command.

--WARNING CL 9580-- Internal formatter error - {text}.

Condition Identifier: CLE\$INTERNAL_FORMATTER_ERROR Code: CL 9580

Description: The SCL formatter encountered a problem that was internal to the formatter itself.

User Action: Write a PSR and include the appropriate support documentation.

--ERROR CL 9600-- Maximum number of errors ({text}) encountered.

Condition Identifier: CLE\$MAX_ERROR_COUNT_REACHED Code: CL 9600

Description: The SCL formatter found enough errors in its input to assume that input is not SCL text. It therefore stops processing input.

User Action: Check that the input is indeed SCL commands.

--ERROR CL 9620-- PAGE_WIDTH value must be at least {text} greater than INITIAL_IDENT_COLUMN value.

Condition Identifier: CLE\$PAGE_WIDTH_TOO_SMALL Code: CL 9620

Description: The SCL formatter requires at least the stated number of columns to work with on the output file.

User Action: Use an output file with a larger page_width or select a smaller indentation.

--ERROR CL 9640-- Utility {text} count exceeded.

Condition Identifier: CLE\$TOO_MANY_NAMES Code: CL 9640

Description: Too many utility command names were given to the SCL formatter via the utility definition file.

User Action: Reduce the number of utility command names.

--ERROR CL 9660-- Utility name {text} same as terminator name.

Condition Identifier: CLE\$UTILITY_SAME_AS_END Code: CL 9660

Description: The names that begin and end a utility must be different.

User Action: Correct the utility definition.

--WARNING CL 9680-- {text} Warning messages.

Condition Identifier: CLE\$WARNINGS_ENCOUNTERED Code: CL 9680

Description: A command found the stated number of warnings in its input.

User Action: Correct the input for the command.

--ERROR CM 100-- Device file {text} is currently being updated in {text}.

Condition Identifier: CME\$ADF_DEV_FILE_BEING_UPDATED Code: CM 100

Description: A configuration management device file is being updated.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--INFORMATIVE CM 120-- A device file of type {text} with status {text} does not currently exist.

Condition Identifier: CME\$ADF_DEV_FILE_NOT_AVAILABLE Code: CM 120

Description: A configuration management device file cannot be found.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--ERROR CM 130-- Device file version {text} is not the current version of this system.

Condition Identifier: CME\$ADF_ILLEGAL_VERSION Code: CM 130

Description: The configuration management device file on disk does not match the system version.

User Action: Do an installation deadstart.

--ERROR CM 140-- Modifications to device file {text} are in an incomplete state.

Condition Identifier: CME\$ADF_INCOMPLETE_DEVICE_FILE Code: CM 140

Description: The information in the configuration management device file cannot be interpreted.

User Action: Do an installation deadstart.

--ERROR CM 150-- A NIL sequence pointer was encountered with a {text} in {text}.

Condition Identifier: CME\$ADF_NIL_ELEMENT_POINTER Code: CM 150

Description: Information in the given file cannot be read.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--ERROR CM 160-- The logical configuration table is not complete at this point during deadstart.

Condition Identifier: CME\$ADF_NIL_LOGICAL_CONF Code: CM 160

Description: The logical configuration table was not built during deadstart.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--ERROR CM 170-- The physical configuration table is not complete at this point during deadstart.

Condition Identifier: CME\$ADF_NIL_PHYSICAL_CONF Code: CM 170

Description: The physical configuration table was not built during deadstart.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--ERROR CM 172-- {text}.

Condition Identifier: CME\$ADF_NIL_STATE_INFO_TABLE Code: CM 172

Description: The state information table was not built during deadstart.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be required.

--ERROR CM 200-- {text}.

Condition Identifier: CMESIT_INVALID_PARAMETER Code: CM 200

Description: An invalid parameter has been detected in an internal interface.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be necessary.

--ERROR CM 205-- {text} for logical {text} in procedure {text}.

Condition Identifier: CMESIT_LU_UNKNOWN_PRODUCT_ID Code: CM 205

Description: System does not recognize the product identification for a storage device.

User Action: Check the physical configuration definition subcommands that define the installed configuration to ensure that all product identifications are correct. Edit the physical configuration and reinstall it.

--ERROR CM 220-- Element {text} on channel {text} is not a controller.

Condition Identifier: CMESIT_PP_NOT_CONTROLLER_TYPE Code: CM 220

Description: The element specified must be a controller in order to be connected to a channel.

User Action: The element identification of a controller is probably incorrectly specified. Edit the physical configuration and reinstall it.

--ERROR CM 225-- Controller element {text} has mismatching equipment numbers: {text} and {text} in procedure {text}.

Condition Identifier: CMESIT_PP_CT_NUMBER_MISMATCH Code: CM 225

Description: An internal table mismatch has occurred.

User Action: Check the iou_connection parameter in the controller's element definition to ensure all equipment addresses are the same. Edit the physical configuration and reinstall it.

--ERROR CM 230-- {text} not found in controller connection in procedure {text}.

Condition Identifier: CMESIT_PP_CHANNEL_NAME_MISMATCH Code: CM 230

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be necessary.

--ERROR CM 235-- {text} is not a storage device element in procedure {text}.

Condition Identifier: CMESIT_PP_NOT_DATA_TYPE Code: CM 235

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 240-- {text} has an unknown product identifier in procedure {text}.

Condition Identifier: CMESIT_PP_UNKNOWN_PRODUCT_ID Code: CM 240

Description: An unknown product identification has been detected.

User Action: Check the element identification on the physical configuration definition subcommands for correctness. Edit the physical configuration and reinstall it.

--ERROR CM 255-- Physical unit {text} for element {text} does not match defined units in procedure {text}.

Condition Identifier: CMESIT_PP_UNIT_NUMBER_MISMATCH Code: CM 255

Description: A mismatch of internal tables has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 260-- Unsupported channel {text} for element {text} was detected in procedure {text}.

Condition Identifier: CME\$IT_PP_INVALID_CHANNEL Code: CM 260

Description: An incorrect channel number was encountered.

User Action: Check your installed physical configuration to ensure that all channel numbers are correct.

--ERROR CM 270-- {text}.

Condition Identifier: CME\$IT_INVALID_JSN Code: CM 270

Description: An invalid job sequence number was used in the request.

User Action: Reenter the request.

--ERROR CM 275-- {text}.

Condition Identifier: CME\$IT_LUN_NOT_IN_RANGE Code: CM 275

Description: The logical unit number is larger than the number of drives defined in the configuration.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, contact a site analyst; an installation deadstart may be necessary.

--ERROR CM 280-- {text}.

Condition Identifier: CME\$IT_NIL_LUT Code: CM 280

Description: The pointer to the logical unit tables has not been initialized.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 285-- {text}.

Condition Identifier: CME\$IT_UNIT_UNAVAILABLE Code: CM 285

Description: The requested unit is either currently assigned, or may not be assigned to a job.

User Action: Reenter the request at a later time.

--ERROR CM 287-- {text}.

Condition Identifier: CME\$IT_NIL_LPT Code: CM 287

Description: The pointer to the logical PP table has not been initialized.

User Action: Do a continuation deadstart of NOS/VE. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 310-- Element size mismatch: {text}.

Condition Identifier: CME\$JTD_ELEMENT_SIZE_MISMATCH Code: CM 310

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 320-- Eoi not returned: {text}.

Condition Identifier: CME\$JTD_EOI_NOT_RETURNED Code: CM 320

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 330-- Nil sequence pointer: {text}.

Condition Identifier: CME\$JTD_NIL_SEQUENCE_POINTER Code: CM 330

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be necessary.

--ERROR CM 340-- System device logical unit number not found.

Condition Identifier: CME\$JTD_SYS_DEV_LUN_NOT_FOUND Code: CM 340

Description: The installed configuration does not include the same path to the system device as defined by the deadstart boot display.

User Action: Check the path to the system device and compare it to the physical configuration file. If the installed configuration is in error, install a corrected configuration. If the installed configuration is correct, redeadstart the system and change the deadstart boot display to correctly define the system device.

--ERROR CM 348-- Unable to set lock on logical unit table.

Condition Identifier: CME\$JTD_UNABLE_TO_ACCESS_LUT Code: CM 348

Description: The logical unit table is currently being accessed by another task or job.

User Action: Retry the state change request later.

--ERROR CM 349-- {text} is already assigned to {text}.

Condition Identifier: CME\$JTD_UNIT_ALREADY_ASSIGNED Code: CM 349

Description: The current element is being assigned to a job.

User Action: Retry the state change request later.

--ERROR CM 350-- {text} is already reserved to {text}.

Condition Identifier: CME\$JTD_UNIT_ALREADY_RESERVED Code: CM 350

Description: The current element is being reserved to a job.

User Action: Retry the state change request later.

--WARNING CM 352-- Unable to update state information, state change may not be retained across deadstart.

Condition Identifier: CME\$JTD_STATE_NOT_RETAINED Code: CM 352

Description: The new state of the element cannot be updated on the device file or, in the case of channel, in the mainframe reconfiguration table.

User Action: The state of the element should be changed manually using PCU or LCU at the next continuation deadstart.

--ERROR CM 400-- Element {text} has already been defined.

Condition Identifier: CME\$LCM_DUPLICATE_ELEMENT_NAMES Code: CM 400

Description: The requested element name is already being used in the physical configuration.

User Action: Reenter your request with a different element name.

--ERROR CM 410-- The active configuration is empty

Condition Identifier: CME\$LCM_EMPTY_LC Code: CM 410

Description: At this time in an installation deadstart, the logical configuration is not active.

User Action: Wait until the installation deadstart is complete. Then you can display the active logical configuration.

--ERROR CM 430-- Permission not granted for : {text}.

Condition Identifier: CME\$LCM_RING_VALIDATION_ERROR Code: CM 430

Description: User is attempting to use an interface or command with insufficient ring privileges.

User Action: Consult site analyst or family administrator.

--ERROR CM 431-- User is executing in the wrong job class to change the element state from {text} to {text}.

Condition Identifier: CME\$JOBCLASS_CANT_CHANGE_STATES Code: CM 431

Description: The job class of the job which executed this command is not validated to change the element states.

User Action: Consult site analyst or family administrator.

--ERROR CM 432-- User cannot change the element state from {text} to {text} because the element is in its current state due to hardware problems.

Condition Identifier: CME\$CANNOT_CHANGE_STATES Code: CM 432

Description: The element was placed in its current state due to system-detected hardware problems, and it cannot change states until these problems have been fixed.

User Action: Not applicable.

--ERROR CM 433-- The state of the element changed from {text} to {text} between the initial validation of the element state and the actual request to change the element state.

Condition Identifier: CME\$STATE_CHANGED_IN_INTERIM Code: CM 433

Description: In the interval time between the initial validation of the element state change and the actual request to change the element state, the element underwent a state change which made the initial state change request invalid.

User Action: Not applicable.

--ERROR CM 434-- The element state change of {text} to {text} would produce a system interrupt.

Condition Identifier: CME\$CHANGE_PRODUCES_INTERRUPT Code: CM 434

Description: Any change of state for the specified element will cause an interruption of the system operations.

User Action: Do not attempt to change this element's state.

--ERROR CM 435-- Empty physical address specifier set.

Condition Identifier: CME\$LCM_EMPTY_PA_SET Code: CM 435

Description: The CYBIL type specified an element that could not be found in the active configuration.

User Action: Make sure that the CYBIL type specifies an element in the active configuration.

--ERROR CM 436-- The state change from {text} to {text} is not supported.

Condition Identifier: CME\$LCM_UNSUPPORTED_STATECHANGE Code: CM 436

Description: The element cannot change states in such a manner because the requested state change is not currently supported by NOS/VE.

User Action: Do not attempt to change this element's state.

--ERROR CM 437-- The state change from {text} to {text} is not supported for this processor model.

Condition Identifier: CME\$LCM_MODEL_CANT_CHANGE_STATE Code: CM 437

Description: The model of this element cannot change to the requested state.

User Action: Do not attempt to change the element to the requested state.

--ERROR CM 438-- User cannot change the element state from {text} to {text} because the element was in the current state at CTI deadstart.

Condition Identifier: CME\$CANT_CHANGE_STATE_WO_DEADST Code: CM 438

Description: The state of this element was set at CTI deadstart time and cannot be changed.

User Action: Change the element state at the CTI deadstart and then perform a continuation deadstart.

--ERROR CM 440-- {text} missing in physical address specifier set.

Condition Identifier: CME\$LCM_MISSING_PA_SET_MEMBER Code: CM 440

Description: The CYBIL type is missing one of the three values.

User Action: Be sure the CYBIL type specifies all three values.

--ERROR CM 445-- Element {text} is not in the active configuration.

Condition Identifier: CME\$LCM_ELEMENT_NOT_FOUND Code: CM 445

Description: The element name specified is not in the active configuration.

User Action: Check the active configuration and reenter the correct name.

--WARNING CM 446-- The current element state is the same as the requested state.

Condition Identifier: CME\$REQUEST_STATE_IS_CRNT_STATE Code: CM 446

Description: The element is already in the requested state; for example, there would be no actual state change.

User Action: Check the element state and/or the element state which you wish to change.

--ERROR CM 450-- An element name cannot be found for physical address: {text}.

Condition Identifier: CME\$LCM_ELEMENT_NAME_NOT_FOUND Code: CM 450

Description: An element name cannot be found in the active configuration for the specified physical address.

User Action: Check the logical configuration subcommand and ensure physical addresses are correct.

--ERROR CM 451-- Hardware address must be specified for multi-spindles devices.

Condition Identifier: CME\$LCM_AMBIGUOUS_PRODUCT_ID Code: CM 451

Description: The given product ID and serial number cannot be used to determine the element name because serial number uniqueness is not enforced for multispindle devices.

User Action: Specify hardware address to query element name.

--ERROR CM 456-- {text} is a system critical element.

Condition Identifier: CME\$LCM_SYSTEM_CRITICAL_ELEMENT Code: CM 456

Description: The element is either the system device, the only controller or channel to the system device, or the only CPU on the mainframe.

User Action: Check the active configuration, and retry the request on a different element.

--ERROR CM 459-- {text} requires channel port on IOU_CONNECTIONS parameter.

Condition Identifier: CME\$CHANNEL_PORT_REQUIRED Code: CM 459

Description: The current device must be configured on a CIO channel with port.

User Action: Edit the physical configuration and change the IOU_CONNECTION to use the CIO channel with port.

--ERROR CM 510-- File inconsistency in the {text} detected in procedure {text}.

Condition Identifier: CME\$LCU_ELEMENT_SIZE_MISMATCH Code: CM 510

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 520-- Information regarding the configuration is not available.

Condition Identifier: CME\$LCU_EOI_NOT_RETURNED Code: CM 520

Description: An internal table mismatch has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 530-- File {file} must have page width of at least {text} characters.

Condition Identifier: CME\$LCU_ILLEGAL_OUTPUT_FILE Code: CM 530

Description: Page width for the output file is too small.

User Action: Change the file attributes to have a page width of at least 60 characters.

--ERROR CM 535-- Parameter : {text} is not a legal value for the state field of the CONFIGURATION_OPTION parameter.

Condition Identifier: CME\$LCU_ILLEGAL_STATE_FIELD Code: CM 535

Description: The CONFIGURATION_OPTION parameter of the DISPLAY_ELEMENT subcommand is not correct.

User Action: Check the description of the LCU DISPLAY_ELEMENT subcommand and retry the subcommand with a correct parameter.

--ERROR CM 536-- Parameter : {text} is not a legal value for the type field of the CONFIGURATION_OPTION parameter.

Condition Identifier: CME\$LCU_ILLEGAL_TYPE_FIELD Code: CM 536

Description: The DISPLAY_OPTION parameter of the DISPLAY_ELEMENT subcommand is not correct.

User Action: Check the description of the LCU DISPLAY_ELEMENT subcommand and retry the subcommand with a correct parameter.

--ERROR CM 540-- Illegal value for parameter {text}.

Condition Identifier: CME\$LCU_IMPROPER_NAME_VALUE Code: CM 540

Description: The parameter specified is not correct.

User Action: Check the description of the INSTALL_LOGICAL_CONFIGURATION subcommand and retry the subcommand with a correct value.

--ERROR CM 541-- Illegal initialization of system device {text}.

Condition Identifier: CME\$LCU_ILLEGAL_INIMV Code: CM 541

Description: An attempt was made to initialize the system device when it was already initialized.

User Action: Check the element name on the INITIALIZE_MS_VOLUME command to ensure that it is not the system device.

--ERROR CM 542-- {text} is already used for the system device.

Condition Identifier: CME\$LCU_DUPLICATE_VSN Code: CM 542

Description: An attempt was made to use a VSN already assigned to the system device.

User Action: Choose a different VSN.

--ERROR CM 543-- Illegal ADDVTS issued for system device element.

Condition Identifier: CME\$LCU_ILLEGAL_ADDVTS Code: CM 543

Description: An attempt was made to add the system device to the set.

User Action: Check the VSN on the ADD_VOLUME_TO_SET command and ensure that it is not the system device recorded vsn.

--ERROR CM 550-- VSN parameter: {text} contains non alphanumeric characters.

Condition Identifier: CME\$LCU_INVALID_VSN Code: CM 550

Description: The recorded VSN contains nonalphanumeric characters.

User Action: Retry the subcommand using a valid VSN.

--ERROR CM 560-- Nil sequence pointer: {text}.

Condition Identifier: CME\$LCU_NIL_SEQUENCE_POINTER Code: CM 560

Description: An internal file is not in a stable state.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 561-- Asynchronous attempts not allowed for : {text}.

Condition Identifier: CME\$LCU_ASYNCHRONOUS_ATTEMPT Code: CM 561

Description: There are two or more asynchronous tasks attempting to change the mass storage classes in the configuration.

User Action: Make sure that only one task is doing the mass storage class changes.

--ERROR CM 562-- {text} is NOT enabled.

Condition Identifier: CME\$LCU_INVALID_ELEMENT_STATUS Code: CM 562

Description: The mass storage device is currently disabled, or all paths to the elements are not in the ON state.

User Action: Contact a site analyst.

--ERROR CM 564-- Cannot remove class A from any devices.

Condition Identifier: CME\$LCU_CANNOT_DELETE_CLASS_A Code: CM 564

Description: An attempt was made to delete class A from the mass storage device.

User Action: Class A must reside on all devices. Do not delete class A with the CHANGE_MS_CLASS subcommand.

--ERROR CM 565-- Cannot remove class A or Q from the system device.

Condition Identifier: CME\$LCU_INVALID_SYS_DEV_CLASS Code: CM 565

Description: An attempt was made to delete class A or Q from the system device.

User Action: Do not delete class A or Q from the system device.

--FATAL CM 566-- The following classes have no members : {text}. You must use the CHANGE_MASS_STORAGE_CLASS command to add those classes back.

Condition Identifier: CME\$LCU_CLASS_HAS_NO_MEMBERS Code: CM 566

Description: Some mass storage classes do not exist in the configuration.

User Action: Use the LCU command CHANGE_MS_CLASS to add those classes in.

--ERROR CM 569-- The DEFINE_HOST_NETWORK command can only be specified once.

Condition Identifier: CME\$LCU_DUP_HOST_NETWORK_DEFN Code: CM 569

Description: The network configuration contains multiple definitions of the host network identifier. Only one definition is allowed.

User Action: Remove duplicate host network definitions from the \$SYSTEM.NETWORK.CONFIGURATION.

--ERROR CM 570-- Push returned nil pointer.

Condition Identifier: CME\$LCU_PUSH_FAILED Code: CM 570

Description: A push instruction returned a NIL pointer. This indicates a potential memory problem.

User Action: Exit the Logical Configuration Utility. Reenter and try again.

--ERROR CM 571-- System id {text} has already been assigned.

Condition Identifier: CME\$LCU_DUPLICATE_SYSTEM_ID Code: CM 571

Description: The same system identifier has been specified for two OSI mode ICA-II's via the DEFINE_NETWORK_CONNECTION subcommand. Each ICA-II must have a unique system identifier specified.

User Action: Obtain the correct system identifiers from the customer engineer and modify the network configuration accordingly.

--ERROR CM 572-- The network configuration must define a network device.

Condition Identifier: CME\$LCU_NO_NETW_DEVICE_DEFN Code: CM 572

Description: The network configuration does not contain a network device definition.

User Action: Add the appropriate network device definition to the network configuration.

--WARNING CM 575-- System/Deadstart device path of {text}, {text}, EQUIPMENT {text}, and UNIT {text} is not in the configuration file, {text}.

Condition Identifier: CME\$LCU_SYS_DEV_PATH_NOT_FOUND Code: CM 575

Description: The physical path to the system device defined by system core commands and the SETVE procedure does not match any unit in the configuration.

User Action: Check the installed physical configuration and make sure that the same physical path is displayed on the deadstart boot display.

--ERROR CM 576-- Too many values for the CONFIGURATION_OPTIONS parameter.

Condition Identifier: CME\$LCU_TOO_MANY_CONF_OPTIONS Code: CM 576

Description: There are too many values given for the CONFIGURATION_OPTION parameter on the DISPLAY_ELEMENT subcommand.

User Action: Retry the subcommand with a valid number of parameters.

--ERROR CM 577-- Requested element {text} does not support the requested density.

Condition Identifier: CME\$LCU_UNIT_DENSITY_MISMATCH Code: CM 577

Description: The density specified on the TYPE parameter of the INITIALIZE_TAPE_VOLUME command is not supported on the tape drive specified by the ELEMENT_NAME parameter of the INITIALIZE_TAPE_VOLUME command.

User Action: Specify an ELEMENT_NAME that supports the desired density or change the density specified by the TYPE parameter.

--ERROR CM 578-- The requested element {text} is not a tape unit.

Condition Identifier: CME\$LCU_ILLEGAL_TAPE_UNIT_NAME Code: CM 578

Description: The ELEMENT_NAME specified on the INITIALIZE_TAPE_VOLUME command is not the element name of a NOS/VE tape unit.

User Action: Specify the element name of a NOS/VE tape unit on the INITIALIZE_TAPE_VOLUME command.

--ERROR CM 579-- The wrong tape element, {text}, was assigned. The requested element name was {text}.

Condition Identifier: CME\$LCU_WRONG_ELEMENT_ASSIGNED Code: CM 579

Description: The ELEMENT_NAME specified on the ASSIGN_DEVICE command is not the same as the ELEMENT_NAME specified on the INITIALIZE_TAPE_VOLUME command.

User Action: Specify the same ELEMENT_NAME on the ASSIGN_DEVICE command as is specified on the INITIALIZE_TAPE_VOLUME command.

--ERROR CM 580-- Network configuration requires DEFINE_HOST_NETWORK command.

Condition Identifier: CME\$LCU_NO_HOST_NETWORK_DEFN Code: CM 580

Description: The network configuration does not contain a definition for the host network identifier.

User Action: Add a DEFINE_HOST_NETWORK subcommand to the network configuration file.

--WARNING CM 582-- {text} will be flawed when available.

Condition Identifier: CME\$LCU_SAVE_FLAW_WARNING Code: CM 582

Description: The LCU subcommand DEFINE_MS_FLAW cannot apply the entered flaw at this time. The flaw will be saved until the device is available and the flaw will then be applied.

User Action: None.

--ERROR CM 583-- {text}

Condition Identifier: CME\$LCU_ILLEGAL_PARAMETER_VALUE Code: CM 583

Description: You have attempted to use the wrong value kind in an LCU subcommand parameter.

User Action: Check your LCU subcommands and make sure the parameter values are proper.

--WARNING CM 584-- Address not software flawed - {text}.

Condition Identifier: CME\$LCU_ADDRESS_NOT_SW_FLAWED Code: CM 584

Description: The lcu subcommand REMOVE_MS_FLAW found that the address given was not flawed.

User Action: Re-enter the correct address.

--ERROR CM 585-- {text}

Condition Identifier: CME\$LCU_CHANNEL_NUMBER_TOO_BIG Code: CM 585

Description: The channel number value on the command is out of range.

User Action: Check the commands in your network prolog.

--ERROR CM 586-- {text} is already defined.

Condition Identifier: CME\$LCU_DUPLICATE_CHANNELS Code: CM 586

Description: One of the channels specified on the LCU subcommand DEFINE_CHANNEL_NETWORK is already being used.

User Action: Check the commands in your network prolog or retry with different values.

--ERROR CM 588-- User has insufficient privilege to install the network configuration.

Condition Identifier: CME\$LCU_UNDER_PRIV_NAM_USER Code: CM 588

Description: User is attempting to install the network configuration while running with insufficient ring privileges.

User Action: Check your input file.

--ERROR CM 589-- Network id {text} has already been assigned.

Condition Identifier: CME\$LCU_DUPLICATE_NETWORK_ID Code: CM 589

Description: The same network identifier has been assigned to two different networks. Each network must have a unique network identifier.

User Action: Check your LCU commands and use a unique value for each network ID.

--ERROR CM 590-- {text} detects {text} error(s).

Condition Identifier: CME\$LCU_VERNC_ERROR Code: CM 590

Description: Either the VERIFY_NETWORK_CONFIGURATION command or the INSTALL_NETWORK_CONFIGURATION command has detected errors in the network configuration.

User Action: Note the specific errors indicted and correct the network configuration.

--ERROR CM 591-- Input file {text} is empty.

Condition Identifier: CME\$LCU_EMPTY_INPUT_ON_VERNC Code: CM 591

Description: The input file on the LCU subcommand VERIFY_NETWORK_CONFIGURATION is empty.

User Action: Not applicable.

--ERROR CM 595-- Cylinder parameter too large - {text}.

Condition Identifier: CME\$LCU_CYLINDER_LIMIT_EXCEEDED Code: CM 595

Description: The CYLINDER parameter entered via the DEFINE_MS_FLAW or the REMOVE_MS_FLAW subcommand was larger than the physical size of the disk.

User Action: Re-enter the subcommand with the correct cylinder.

--ERROR CM 596-- Track parameter too large - {text}.

Condition Identifier: CME\$LCU_TRACK_LIMIT_EXCEEDED Code: CM 596

Description: The TRACK parameter entered via the DEFINE_MS_FLAW or the REMOVE_MS_FLAW subcommand was larger than the physical number of tracks.

User Action: Re-enter the command with the correct value for the TRACK parameter.

--ERROR CM 597-- Sector parameter too large - {text}.

Condition Identifier: CME\$LCU_SECTOR_LIMIT_EXCEEDED Code: CM 597

Description: The SECTOR parameter entered via the DEFINE_MS_FLAW or the REMOVE_MS_FLAW subcommand was larger than the physical number of sectors.

User Action: Re-enter the value for the SECTOR parameter.

--ERROR CM 598-- Logging not active - {text}.

Condition Identifier: CME\$LCU_LOGGING_NOT_ACTIVE Code: CM 598

Description: For the LCU subcommand DEFINE_MS_FLAW to execute, the device must be active (the logger must be running on this device). At this point, the logger is not running and flaws cannot be defined.

User Action: Re-enter the subcommand after adding the volume to the set, or define flaws at system core time.

--WARNING CM 599-- Address already flawed - {text}.

Condition Identifier: CME\$LCU_ADDRESS_ALREADY_FLAWED Code: CM 599

Description: The LCU subcommand DEFINE_MS_FLAW found that the disk address given was already flawed.

User Action: None.

--ERROR CM 605-- Element {text} is not in the physical configuration table.

Condition Identifier: CME\$PC_ELEMENT_NOT_FOUND Code: CM 605

Description: The element specified was not found in the physical configuration table or was in the OFF or DOWN state.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 610-- The system device is not configured in the logical configuration.

Condition Identifier: CME\$PC_EQUIP_NOT_CONFIGURED Code: CM 610

Description: The physical path to the system device defined in the installed physical configuration does not match the physical path defined by the deadstart boot display.

User Action: Check the physical path including the channel number, equipment number (usually zero), and the physical unit number of the system device in the installed physical configuration and the deadstart boot display. They all must agree.

--ERROR CM 615-- Element {text} is not a data storage device.

Condition Identifier: CME\$PC_NAME_NOT_DATA_ELEMENT Code: CM 615

Description: The specified element is not a storage device.

User Action: Retry the request with the correct element name.

--ERROR CM 620-- The logical configuration table was found to be empty when looking for element {text}.

Condition Identifier: CME\$PC_NIL_LCT Code: CM 620

Description: The logical configuration table has not been defined at this point in deadstart.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 625-- The physical configuration table was found to be empty when looking for element {text}.

Condition Identifier: CME\$PC_NIL_PCT Code: CM 625

Description: The physical configuration table has not been defined at this point in deadstart.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 630-- No more channels available in the logical configuration.

Condition Identifier: CME\$PC_NO_MORE_CHANNELS Code: CM 630

Description: All channels available have been delivered. If this error is returned to an installer, some mixup in internal tables has occurred.

User Action: Do a continuation deadstart. If the error condition persists, an installation deadstart may be required.

--ERROR CM 635-- Element {text} is not logically configured.

Condition Identifier: CME\$PC_NOT_LOGICALLY_CONF Code: CM 635

Description: The specified element is not part of the active configuration.

User Action: Check the configuration and make sure that the correct name was used.

--ERROR CM 645-- Product identification {text} is unknown to the system.

Condition Identifier: CME\$PC_UNKNOWN_CONTROLLER_TYPE Code: CM 645

Description: An unknown controller product identification has been detected.

User Action: Check the physical configuration file and make sure that the controller product identifications are correct.

--ERROR CM 650-- Channel number {text} is not an external channel.

Condition Identifier: CME\$PC_UNSUPPORTED_CHANNEL Code: CM 650

Description: The channel number specified is not available for use by the system device.

User Action: Redefine the channel to be used for the system device.

--ERROR CM 660-- Recorded vsn {text} contains more than 6 characters.

Condition Identifier: CME\$CMD_NAME_TOO_LARGE Code: CM 660

Description: A vsn of more than six characters was specified.

User Action: Supply a vsn of six or fewer characters.

--ERROR CM 663-- Invalid iou number : {text}.

Condition Identifier: CME\$INVALID_IOU_NUMBER Code: CM 663

Description: An invalid iou number has been encountered internally.

User Action: Contact a site analyst.

--ERROR CM 664-- {text} is not a valid IOU name.

Condition Identifier: CME\$INVALID_IOU_NAME Code: CM 664

Description: The iou name specified on the command/program interface is invalid.

User Action: Check the iou name value used on the command/program interface.

--ERROR CM 666-- {text}.

Condition Identifier: CME\$INVALID_CHANNEL_NUMBER Code: CM 666

Description: The channel number is out of range.

User Action: Check the channel number being used and make sure it is in the proper range.

--ERROR CM 667-- {text} is not a valid channel name.

Condition Identifier: CME\$INVALID_CHANNEL_NAME Code: CM 667

Description: The channel name used is not a valid channel name.

User Action: Check the channel name used, and make sure it has the form CHx, CCHx or CCHxp, where x is an integer value and p is either A or B.

--ERROR CM 685-- Product identification {text} is unknown to the system.

Condition Identifier: CME\$UNKNOWN_PRODUCT_ID Code: CM 685

Description: The product identifier specified on system core command is unknown to the system.

User Action: Check your system core commands and use supported product identifiers.

--ERROR CM 691-- {text}

Condition Identifier: CME\$BOOT_TAPE_IO_ERROR Code: CM 691

Description: A tape I/O error, such as tape not ready or channel parity error, has occurred while validating the deadstart device.

User Action: Check the tape drive for unit ready or consult the site analyst.

--ERROR CM 692-- {text}

Condition Identifier: CME\$BOOT_DISK_IO_ERROR Code: CM 692

Description: A disk I/O error has occurred while reading the system disk.

User Action: Consult site analyst.

--ERROR CM 694-- {text}.

Condition Identifier: CME\$UNABLE_TO_IDLE Code: CM 694

Description: Unable to send idle command to the PP.

User Action: Retry the request. If it fails, try to hardware idle the PP.

--ERROR CM 695-- {text}.

Condition Identifier: CME\$UNABLE_TO_RESUME Code: CM 695

Description: Unable to send resume command to the PP.

User Action: Retry the request. If it fails, consult the site analyst.

--ERROR CM 696-- PP interface table not built.

Condition Identifier: CME\$PPIT_NOT_BUILT Code: CM 696

Description: The PP interface table has not been initialized.

User Action: Consult a site analyst.

--ERROR CM 697-- Hardware idle required for dumping PP memory.

Condition Identifier: CME\$DUMP_REQUIRES_HARDWARE_IDLE Code: CM 697

Description: User has attempted to get a dump of the PP memory without specifying the hardware_idle flag.

User Action: Check the parameters passed to CMP\$IDLE_PP.

--ERROR CM 700-- Element {text} has a physical equipment number, {text} that is already being used by {text}.

Condition Identifier: CME\$PCU_DUPLICATE_PEN Code: CM 700

Description: Attempted to assign a channel to more than one controller.

User Action: Execute the Physical Configuration Utility again and assign a channel to only one controller.

--ERROR CM 710-- Physical unit number {text} of element {text} is already being used by {text}.

Condition Identifier: CME\$PCU_DUPLICATE_PUN Code: CM 710

Description: Attempted to use the same unit number for two elements that have the same controller.

User Action: Execute the Physical Configuration Utility again and, for a tape or disk controller, make the unit numbers unique.

--ERROR CM 715-- Element {text} has a serial number {text} that is already being used.

Condition Identifier: CME\$PCU_DUPLICATE_SER_NUM Code: CM 715

Description: A serial number has already been defined.

User Action: Check your PCU commands and make sure a unique serial number is associated with each element.

--ERROR CM 720-- Information regarding the configuration is not available in the PHYSICAL_CONFIGURATION_UTILITY.

Condition Identifier: CME\$PCU_EOI_NOT_RETURNED Code: CM 720

Description: Missing definitions of elements necessary in a physical configuration.

User Action: Check that the physical configuration subcommands define the channels, controllers, disk units, and tape units that NOS/VE will use.

--ERROR CM 721-- INSTALL_PHYSICAL_CONFIGURATION cannot be called after deadstart.

Condition Identifier: CMESPCU_INVALID_INSPC_CALL Code: CM 721

Description: You are trying to install the configuration after the configuration has been activated.

User Action: Install the configuration only during deadstart.

--ERROR CM 722-- {text} is not yet defined.

Condition Identifier: CMESPCU_SAME_AS_NOT_DEFINED Code: CM 722

Description: You are trying to reference an element through the SAME_AS parameter while the element is not yet defined.

User Action: Define the element first or do not use the SAME_AS parameter.

--ERROR CM 730-- Can't detect eoi for input file in {text} in the PHYSICAL_CONFIGURATION_UTILITY.

Condition Identifier: CMESPCU_FILE_ERROR Code: CM 730

Description: Detected an internal file inconsistency during the physical configuration utility session.

User Action: Contact Control Data central software support.

--ERROR CM 735-- File {file} must have page width of at least {text} characters.

Condition Identifier: CMESPCU_ILLEGAL_OUTPUT_FILE Code: CM 735

Description: Page width for the output file is too small.

User Action: Specify a page width of at least 60 characters for the output file.

--ERROR CM 740-- Invalid channel name: {text} in {text}.

Condition Identifier: CMESPCU_INVALID_CHANNEL_NUMBER Code: CM 740

Description: Specified an incorrect integer for the CHANNEL_NUMBER parameter in the SET_DATA_CHANNEL_DEFINITION.

User Action: Correct the integer on the CHANNEL_NUMBER parameter and reexecute the Physical Configuration Utility.

--ERROR CM 745-- {text} has an invalid PRODUCT_IDENTIFICATION parameter: {text}.

Condition Identifier: CMESPCU_INVALID_PRODUCT_ID Code: CM 745

Description: The product identifier for an element is incorrect.

User Action: Correct the error by adding a \$ to the product identification and reexecute the Physical Configuration Utility.

--ERROR CM 755-- {text} for element {text}.

Condition Identifier: CMESPCU_INVALID_PARAMETER Code: CM 755

Description: Specified a parameter that the Physical Configuration Utility does not recognize.

User Action: Correct the error (check the spelling of the parameter) and reexecute the Physical Configuration Utility.

--ERROR CM 760-- {text} for element {text}.

Condition Identifier: CMESPCU_TOO_MANY_VALUE_SETS Code: CM 760

Description: Specified too many value sets in the parameter list.

User Action: Shorten the parameter list and reexecute the Physical Configuration Utility.

--ERROR CM 765-- User has insufficient privilege to install the physical configuration.

Condition Identifier: CMESPCU_UNDERPRIVILEGED_USER Code: CM 765

Description: In order to install the physical configuration, the user must be at the system console and run under the system job.

User Action: Use the VERIFY_PHYSICAL_CONFIGURATION command to verify the configuration file.

--INFORMATIVE CM 770-- Product identification {text} is unknown to the system.

Condition Identifier: CME\$PCU_UNKNOWN_PRODUCT_ID Code: CM 770

Description: The product ID specified on the PCU command is unsupported.

User Action: Check your commands and use a supported product identifier.

--ERROR CM 775-- {text}: {text}.

Condition Identifier: CME\$PCU_UNKNOWN_VALUE Code: CM 775

Description: Specified an incorrect product identification in defining the element.

User Action: Specify the correct product identification and reexecute the Physical Configuration Utility.

--ERROR CM 776-- {text} is not in the edited file.

Condition Identifier: CME\$PCU_ELEMENT_NOT_FOUND Code: CM 776

Description: The element specified is not in the current file.

User Action: Check the element name used.

--ERROR CM 777-- The current file is empty.

Condition Identifier: CME\$PCU_EMPTY_FILE Code: CM 777

Description: The physical configuration file is empty. You cannot display any elements.

User Action: Only display or change elements when the file is not empty.

--ERROR CM 778-- {text} does not have a connection of the type described.

Condition Identifier: CME\$PCU_CONNECTION_NOT_FOUND Code: CM 778

Description: One of the connection parameters specified cannot be found. This occurs when trying to change the IOU connection of an element that does not have one.

User Action: Display the element with the DISPLAY_ELEMENT_DEFINITION command and retry with the proper connection.

--ERROR CM 779-- {text} cannot be executed within EDIPC session.

Condition Identifier: CME\$PCU_INVALID_COMMAND Code: CM 779

Description: The specified commands cannot be executed while in EDIT_PHYSICAL_CONFIGURATION.

User Action: Exit the subutility to execute these commands.

--ERROR CM 780-- {text} in DEFINE_ELEMENT.

Condition Identifier: CME\$PCU_VALUE_OUT_OF_RANGE Code: CM 780

Description: Specifies a parameter value that is too large or too small.

User Action: Specify a value for the parameter that is within its range of values and reexecute the Physical Configuration Utility.

--ERROR CM 781-- Missing parameter {text} for element {text}.

Condition Identifier: CME\$PCU_MISSING_PARAMETERS Code: CM 781

Description: The specified parameter is missing for the indicated element.

User Action: Use the EDIT_PHYSICAL_CONFIGURATION command to check the command for the specified element.

--ERROR CM 782-- VERIFY_PHYSICAL_CONFIGURATION detects errors.

Condition Identifier: CME\$PCU_VERPC_ERR Code: CM 782

Description: Either INSTALL_PHYSICAL_CONFIGURATION or VERIFY_PHYSICAL_CONFIGURATION has detected an error. If INSPC, the configuration is not activated.

User Action: Check your \$ERROR file and use EDIT_PHYSICAL_CONFIGURATION to correct the error.

--ERROR CM 783-- INSTALL_PHYSICAL_CONFIGURATION detects errors.

Condition Identifier: CME\$PCU_INSPC_ERR Code: CM 783

Description: INSTALL_PHYSICAL_CONFIGURATION has detected an error. The configuration is not activated.

User Action: Check the \$ERROR file and use the EDIT_PHYSICAL_CONFIGURATION subcommand to correct the error.

--ERROR CM 784-- User cannot reference permanent files at this time.

Condition Identifier: CME\$PCU_INVALID_PF_REF Code: CM 784

Description: User is trying to reference a permanent file during a continuation or recovery deadstart.

User Action: Use a local file instead of a permanent file.

--ERROR CM 785-- {text} is already defined in the edited file.

Condition Identifier: CME\$PCU_ELEMENT_ALREADY_DEFINED Code: CM 785

Description: The element is already being used in the configuration file.

User Action: Use the CHANGE_ELEMENT_DEFINITION or CHANGE_ELEMENT_NAME command of the EDIT_PHYSICAL_CONFIGURATION subutility to change the name.

--ERROR CM 786-- {text} is not defined in the configuration file.

Condition Identifier: CME\$PCU_MAINFRAME_NOT_FOUND Code: CM 786

Description: The mainframe specified on the VERIFY_PHYSICAL_CONFIGURATION subcommand cannot be found on the physical configuration file, or the default mainframe name is not present on an attempt to install the configuration.

User Action: Use the EDIT_PHYSICAL_CONFIGURATION subcommand to check the physical configuration file for the presence of the mainframe name.

--ERROR CM 787-- Invalid mixture of single and dual PP tape subsystem: {text} is already defined as a dual PP while {text} is a single PP tape subsystem.

Condition Identifier: CME\$PCU_INCONSISTENT_TAPE_SUBS Code: CM 787

Description: The configuration file contains definitions of single and dual pp tape subsystems.

User Action: Only one type of tape subsystem should be used in the configuration. Check your configuration file and redefine all your tape subsystem definitions as single or double pp.

--ERROR CM 788-- Invalid mixture of product identification connected to {text}. The following products are found : {text}, {text} .

Condition Identifier: CME\$INVALID_MIXTURE_OF_PRODUCT Code: CM 788

Description: The configuration file contains definitions of two different peripheral products on the same channel.

User Action: Check the configuration file and ensure that only one peripheral product is defined per channel.

--ERROR CM 789-- RETAIN parameter can be specified only when using ALL on ELEMENTS parameter.

Condition Identifier: CME\$PCU_INVALID_USE_OF_RETAIN Code: CM 789

Description: The RETAIN parameter of the DELETE_ELEMENT_DEFINITION subcommand can only be used when ALL is specified on the ELEMENTS parameter.

User Action: Retry the request without any value specified for the RETAIN parameter.

--ERROR CM 790-- Element : {text} is found to have no units connected to to it, dangling controller definition is not allowed.

Condition Identifier: CME\$PCU_INVALID_CONNECTION Code: CM 790

Description: A controller definition is found to have no units connection.

User Action: Check the configuration file and make sure the controller has a unit(s) connected to it or delete the definition of the controller.

--ERROR CM 900-- {text} is not reservable.

Condition Identifier: CME\$ELEMENT_NOT_RESERVABLE Code: CM 900

Description: The element cannot be reserved by CMP\$RESERVE_ELEMENT.

User Action: Consult a site analyst.

--ERROR CM 901-- Requested element type does not match element type found in active configuration for {text}.

Condition Identifier: CME\$INCORRECT_ELEMENT_TYPE Code: CM 901

Description: The ELEMENT_TYPE in a program interface call did not match the ELEMENT_TYPE of the element in the physical configuration.

User Action: Correct the ELEMENT_TYPE in the call.

--ERROR CM 902-- Insufficient privilege to perform {text} request.

Condition Identifier: CME\$PRIVILEGED_JOB_REQUIRED Code: CM 902

Description: The user needs to have maintenance job class, system job, or ring 6 privileges to use the interfaces.

User Action: Consult a site analyst or the family administrator.

--ERROR CM 904-- A specific pp cannot be reserved at this time.

Condition Identifier: CME\$SPECIFIC_PP_NOT_RESERVABLE Code: CM 904

Description: You cannot reserve a specific PP on a dual state system.

User Action: Retry the request without reserving a specific PP if running on a dual state system.

--ERROR CM 906-- {text} is already reserved by: {text}.

Condition Identifier: CME\$ELEMENT_ALREADY_RESERVED Code: CM 906

Description: The element is already being reserved to another job.

User Action: Retry later.

--ERROR CM 908-- {text} is NOT in ON state.

Condition Identifier: CME\$ELEMENT_STATE_NOT_PROPER Code: CM 908

Description: The element is not in the proper state for the reservation request to complete.

User Action: Check the state of the element and make sure it is ON.

--ERROR CM 910-- {text} is not available in the NOS or NOS/BE system.

Condition Identifier: CME\$ELEMENT_UNAVAILABLE_IN_170 Code: CM 910

Description: The element cannot be acquired from the NOS or NOS/BE system.

User Action: Check your NOS or NOS/BE configuration for the path to the element.

--ERROR CM 912-- {text} needs to be in the DOWN or OFF state in mainframe {text}.

Condition Identifier: CME\$INVALID_STATE_IN_MAINFRAME Code: CM 912

Description: The element is described as being connected to another mainframe, but it is not DOWN or OFF.

User Action: Check the state of the element on the other mainframe and retry it when it is OFF or DOWN.

--ERROR CM 914-- {text} is not reserved to this job.

Condition Identifier: CME\$ELEMENT_NOT_RESERVED Code: CM 914

Description: The element is not reserved by the job.

User Action: Try to reserve the element before doing something such as CMP\$EXECUTE_PP_PROGRAM.

--ERROR CM 916-- Job already has {text} reserved to it; reservation of {text} not permitted.

Condition Identifier: CME\$RESERVE_NOT_PERMITTED Code: CM 916

Description: Reservation of the element is not permitted to the job because of job class.

User Action: Change the job class.

--ERROR CM 918-- {text} has at least one element connected to it which is not in the OFF state.

Condition Identifier: CME\$ELEMENT_DOWNLINE_CONNECTED Code: CM 918

Description: The element cannot be reserved because it has active downline connected elements.

User Action: Use the LCU subcommand CHANGE_ELEMENT_STATE to change the state of all active downline connected elements to OFF.

--ERROR CM 922-- Active physical configuration table is empty.

Condition Identifier: CME\$ACTIVE_PC_EMPTY Code: CM 922

Description: The active physical configuration table has not been built during deadstart.

User Action: Consult a site analyst.

--ERROR CM 926-- CMP\$EXECUTE_PP_PROGRAM allows no more than two program descriptions.

Condition Identifier: CME\$TOO_MANY_PP_PROGRAM_DESC Code: CM 926

Description: You specified more than 2 entries for the PP_PROGRAM_DESCRIPTION parameter on CMP\$EXECUTE_PP_PROGRAM.

User Action: Specify only 1 or 2 entries for PP_PROGRAM_DESCRIPTION.

--WARNING CY 0-- *CYBIL* , {text}

Condition Identifier: CYE\$CYBIL_WARN_ERROR Code: CY 0

Description: Indicates that the compiler has encountered an irregularity in the source code.

User Action: See the list file. The user may choose to ignore these warnings.

--ERROR CY 1-- *CYBIL* , {text}

Condition Identifier: CYE\$CYBIL_NON_FATAL_ERROR Code: CY 1

Description: A non-fatal error has been diagnosed during compilation.

User Action: See the list file for a list of diagnostics, correct the program, and recompile.

--FATAL CY 2-- *CYBIL FATAL ERROR IN , {text}

Condition Identifier: CYE\$CYBIL_FATAL_ERROR Code: CY 2

Description: Fatal CYBIL errors have been found in the input file.

User Action: See the list file for a list of diagnostics, correct the program, and recompile.

--CATASTROPHIC CY 3-- *CYBIL TERMINATED* , {text}

Condition Identifier: CYE\$CYBIL_ABORT_ERROR Code: CY 3

Description: Compilation has been terminated due to an irrecoverable situation. Basically, the compiler can not continue to process the source code.

User Action: Check the source listing for an indication of the error that has occurred (e.g., array too large). Correct the program and recompile.

--ERROR CY 5000-- NIL pointer to procedure called at P = {text}.

Condition Identifier: CYE\$NIL_CALLED Code: CY 5000

Description: The pointer to the procedure called at the address indicated has a value of NIL.

User Action: Correct the program and recompile.

--ERROR CY 5001-- CYBIL run time error, {text}, detected at line {text} of {text}.

Condition Identifier: CYE\$CYBIL_ABORT Code: CY 5001

Description: An error (e.g., a range error) has occurred during execution of a CYBIL program.

User Action: Correct the program and recompile.

--ERROR CY 6210-- File name too long, {file}.

Condition Identifier: CYE\$FILE_NAME_TOO_LONG Code: CY 6210

Description: This message indicates the incorrect use of a CYBIL I/O routine. The specified file name has more characters than the operating system will allow.

User Action: Refer to the SCL for NOS/VE Language Definition manual for a complete description of NOS/VE naming conventions. Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6215-- File NOT open.

Condition Identifier: CYE\$FILE_NOT_OPEN Code: CY 6215

Description: This message indicates the incorrect use of a CYBIL I/O routine. An undefined variable of type cyt\$file was passed to a CYBIL I/O routine other than CYP\$OPEN_FILE.

User Action: Refer to CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6220-- Incorrect file name, {file}.

Condition Identifier: CYE\$ILLEGAL_FILE_NAME Code: CY 6220

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to open a file with a name that does not conform to the file naming conventions of the operating system.

User Action: Refer to the SCL for NOS/VE Language Definition manual for a complete description of NOS/VE naming conventions. Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6225-- Incorrect input request for {file}.

Condition Identifier: CYE\$ILLEGAL_INPUT_REQUEST Code: CY 6225

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to read from a file that was opened only for output.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6230-- Incorrect display line position for {file}.

Condition Identifier: CYE\$ILLEGAL_LINE_NUMBER Code: CY 6230

Description: This message indicates the incorrect use of a CYBIL I/O routine. A line number less than 1 was passed to the CYP\$POSITION_DISPLAY_PAGE procedure.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6231-- Incorrect tab column for {file}.

Condition Identifier: CYE\$ILLEGAL_TAB_COLUMN Code: CY 6231

Description: This message indicates the incorrect use of a CYBIL I/O routine. A tab_column less than 1 was passed to the CYP\$TAB_FILE procedure.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6235-- Incorrect open request for {file}.

Condition Identifier: CYE\$ILLEGAL_OPEN_REQUEST Code: CY 6235

Description: This message indicates the incorrect use of a CYBIL I/O routine. An invalid combination of parameters was given to an open procedure.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6237-- Incorrect command issued to {file}.

Condition Identifier: CYESILLEGAL_OPERATION Code: CY 6237

Description: This message indicates the incorrect use of a CYBIL I/O routine. An operation was attempted that does not match the FILE_KIND specified for the file on the call to the open file procedure. For example, a CYP\$GET_NEXT_BINARY may have been attempted on a file opened as a text file.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6240-- Incorrect output request for {file}.

Condition Identifier: CYESILLEGAL_OUTPUT_REQUEST Code: CY 6240

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to write to a file that was opened only for input.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6245-- Incorrect skip count {file}.

Condition Identifier: CYESILLEGAL_SKIP_COUNT Code: CY 6245

Description: This message indicates the incorrect use of a CYBIL I/O routine. A skip count less than -1 was passed to the CYP\$SKIP_LINES procedure.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6250-- Key beyond E-O-I on {file}.

Condition Identifier: CYESKEY_PAST_EOI Code: CY 6250

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to perform a binary file operation with a key that was outside the bounds of the file. In other words, the key did not specify a random address that is in the file.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6251-- Premature end of operation on file {file}.

Condition Identifier: CYESPREMATURE_END_OF_OPERATION Code: CY 6251

Description: This message indicates the incorrect use of a CYBIL I/O routine. A boundary condition was encountered during the CYP\$POSITION_RECORD_FILE procedure before the count was exhausted.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6255-- No memory to open file {file}.

Condition Identifier: CYESNO_MEMORY_TO_OPEN_FILE Code: CY 6255

Description: This message indicates the incorrect use of a CYBIL I/O routine. There was insufficient space to allocate the descriptor and/or buffer for the file.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6256-- Could NOT find file {file}.

Condition Identifier: CYESFILE_NOT_FOUND Code: CY 6256

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to open an old file that CYBIL I/O can not find.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR CY 6257-- File ({file}) already exists.

Condition Identifier: CYE\$FILE_ALREADY_EXISTS Code: CY 6257

Description: This message indicates the incorrect use of a CYBIL I/O routine. An attempt was made to open a new file but a file with that name already exists.

User Action: Refer to the CYBIL for NOS/VE Language Definition manual for a complete description of all CYBIL I/O routines. Correct the CYBIL source code and recompile.

--ERROR DB 1-- The current pva was not found in the module table. Re-enter the command specifying the module parameter.

Condition Identifier: DBE\$PVA_NOT_IN_ANY_MODULE Code: DB 1

Description: The current address does not match an address in the module table generated by the loader. It is most likely in a CYBIL runtime routine.

User Action: Re-enter the command specifying the module parameter.

--ERROR DB 2-- The specified address is not in the specified section.

Condition Identifier: DBE\$PVA_NOT_IN_SECTION Code: DB 2

Description: Either the specified address does not match an address in the current section or the specified address could not be found in the named section.

User Action: Check the source listing for the correct address or section name.

--ERROR DB 3-- Module {text} was not found.

Condition Identifier: DBE\$NAMED_MODULE_NOT_FOUND Code: DB 3

Description: The specified module could not be located in the module address table generated by the loader.

User Action: Check the spelling of the module name. The currently active modules can be displayed using the Debug DISPLAY_CALLS command.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 4-- The specified entry point was not found.

Condition Identifier: DBE\$NAMED_ENTRY_POINT_NOT_FOUND Code: DB 4

Description: The spelling of the specified entry point does not match the spelling of an entry point name in the source program.

The entry point may not exist if the module containing the entry point has not been loaded. Also, entry points can be deleted when a program is bound (see the CREATE_MODULE and CHANGE_MODULE_ATTRIBUTES subcommands of the CREATE_OBJECT_LIBRARY command).

User Action: Check the source listing for the spelling of the entry point name.

--ERROR DB 5-- Line numbers are not available.

Condition Identifier: DBE\$NO_LINE_NUMBERS_IN_MODULE Code: DB 5

Description: The source program was compiled without DEBUG_AIDS=DT. Line number tables were not generated; therefore, a line number cannot be specified in the Debug command.

User Action: Either re-compile the source program specifying the DEBUG_AIDS=DT parameter or use module or machine-level addressing.

--ERROR DB 6-- The specified address is not a line number.

Condition Identifier: DBE\$PVA_NOT_LINE_NUMBER Code: DB 6

Description: The specified address does not match any program variable address in the line number table.

User Action: Check the source listing for the correct line number.

--ERROR DB 7-- The line number {text} was not found.

Condition Identifier: DBE\$LINE_NUMBER_NOT_FOUND Code: DB 7

Description: The line number specified by the LINE parameter must correspond to an executable line.

Nonexecutable statements are specification statements (such as type statements, COMMON, DIMENSION, DATA), FORMAT statements, comment lines, or continuation line.

User Action: Change the line number to a number that corresponds to an executable line.

--INFORMATIVE DB 8-- No symbol table exists for module {text}.

Condition Identifier: DB\$NO_SYMBOL_TABLE_IN_MODULE Code: DB 8

Description: The source program was compiled without DEBUG_AIDS=DT. Symbol tables were not generated; therefore, symbolic names cannot be specified in the Debug command.

User Action: Either re-compile the source program specifying the DEBUG=DT parameter or use module or machine-level addressing.

--ERROR DB 9-- Corresponding symbol number in the symbol table could not be found.

Condition Identifier: DB\$SYMBOL_NUMBER_NOT_FOUND Code: DB 9

Description: This is a Debug internal error.

User Action: Report this to your system analyst.

--ERROR DB 13-- A break cannot be set at the specified line.

Condition Identifier: DB\$CANNOT_BREAK_AT_SPECD_LINE Code: DB 13

Description: A break was set at a non-executable line, a comment line, a continuation line, or a line that begins with a ENDIF or IFEND statement.

User Action: Set the break at an executable line or use a different addressing method, such as module/offset addressing.

--ERROR DB 14-- The LINE and PROCEDURE parameters cannot both be specified.

Condition Identifier: DB\$LINE_PARAM_CONFLICTS Code: DB 14

Description: The LINE parameter can only be combined with the MODULE parameter, not the PROCEDURE parameter. Byte-offset addressing must be used if the PROCEDURE parameter is specified.

User Action: Specify either the LINE and MODULE parameters or the PROCEDURE parameter.

--ERROR DB 15-- The LINE parameter may only be combined with the MODULE parameter.

Condition Identifier: DB\$LINE_GOES_WITH_MODULE_ONLY Code: DB 15

Description: The LINE parameter can only be used to identify a line in the module named in the MODULE parameter. The LINE parameter cannot be combined with the PROCEDURE parameter because line numbers are not meaningful within (defined within the scope of) a CYBIL procedure.

User Action: Use the LINE parameter only with the MODULE parameter.

--ERROR DB 16-- More than one base parameter has been specified.

Condition Identifier: DB\$MORE_THAN_ONE_BASE_SPECD Code: DB 16

Description: Only one base parameter can be set on a SET_BREAK or DISPLAY_MEMORY command. Invalid base parameter combinations for the SET_BREAK command are: address and MODULE, address and entry point, MODULE and entry point, and PROCEDURE and entry point. Invalid base parameter combinations for the DISPLAY_MEMORY command are: address and MODULE and address and SECTION.

User Action: Specify only one base parameter.

--ERROR DB 17-- Section {text} is not in the specified module.

Condition Identifier: DB\$SPECD_SECTION_NOT_IN_MODULE Code: DB 17

Description: Either the specified section or module (or both) is misspelled or the specified section is not in the specified module.

User Action: Check the source listing for either the correct spelling of the section and module names or the correct section or module name.

--ERROR DB 18-- Byte offset is beyond the end of the line.

Condition Identifier: DB\$OFFSET_BEYOND_LINE Code: DB 18

Description: The base address plus the byte offset specifies an address that is beyond the end of the line.

User Action: Specify a smaller value for the byte offset.

--ERROR DB 20-- Break point name {text} already exists.

Condition Identifier: DB\$BREAK_NAME_EXISTS Code: DB 20

Description: Each break point must have a unique name.

User Action: Specify a unique name for the break point.

--ERROR DB 21-- MODULE and SECTION must be specified if no ADDRESS is specified.

Condition Identifier: DB\$BASE_PARAMETER_NOT_SPECED Code: DB 21

Description: Either the ADDRESS parameter or the SECTION and MODULE parameters are required so that a base address can be established. Because the ADDRESS parameter was not specified, the SECTION and MODULE parameters must be specified.

User Action: Specify the SECTION and MODULE parameters on the command.

Further Information: See the Debug Usage manual for information about the DISPLAY_MEMORY and SET_BREAK commands.

--ERROR DB 23-- A break event has been selected more than once.

Condition Identifier: DB\$CONDITION_ALREADY_SELECTED Code: DB 23

Description: A break event cannot be selected more than once.

User Action: Remove the second specification of the break event from the command.

--ERROR DB 24-- The maximum number of breaks is already set.

Condition Identifier: DB\$MAX_NUM_BRKPTS_ALREADY_SET Code: DB 24

Description: The maximum number of breaks, which is 64 breaks, is set.

User Action: Delete a break description using the DELETE_BREAK Debug command before you set another break.

--ERROR DB 25-- An invalid address range has been specified.

Condition Identifier: DB\$INVALID_ADDRS_RANGE Code: DB 25

Description: The specified address range references an address range that cannot be referenced through the BYTE_OFFSET and BYTE_COUNT parameters.

User Action: Adjust the values of the BYTE_OFFSET and BYTE_COUNT parameters to reference a valid address range.

--ERROR DB 26-- Invalid break name.

Condition Identifier: DB\$INVALID_BREAK_NAME Code: DB 26

Description: The keyword ALL cannot be used as a break name.

User Action: Change the break name to a word other than ALL.

--ERROR DB 28-- No trap has occurred.

Condition Identifier: DB\$NO_TRAP_HAS_OCCURRED Code: DB 28

Description: The source program has not executed to a point where Debug can gain control.

User Action: Execute the source program using the RUN command to another point in your program.

Further Information: See the Debug Usage manual for information about the Debug RUN command.

--ERROR DB 29-- The specified stack frame does not exist.

Condition Identifier: DB\$INVALID_SF_SPECIFIED Code: DB 29

Description: The value assigned to the START parameter is greater than the number of stack frames that can be displayed. The number of stack frames that can be displayed is the same as the number of calls in the active call chain.

User Action: Use the DISPLAY_CALLS command to see how many calls are in the active call chain and then specify a value for the START parameter that is equal to or smaller than the number of calls.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 33-- A P register value must be specified as integer or 1..12 hex characters.

Condition Identifier: DB\$INVALID_PREG_VALUE Code: DB 33

Description: The value assigned to the VALUE parameter is too large.

User Action: Specify a value within the following ranges:

An integer in the range -7FFFFFFFFF(16) through 7FFFFFFFFF(16), or

A hexadecimal string that contains a maximum of 12 hex digits (blanks do not count); each hex digit corresponds to four bits.

A hexadecimal integer must begin with a 0 digit and end with the radix specifier (16).

--ERROR DB 34-- An A register value must be specified as integer or 1..12 hex characters.

Condition Identifier: DB\$INVALID_AREG_VALUE Code: DB 34

Description: The value assigned to the VALUE parameter is too large.

User Action: Specify a value within the following ranges:

An integer in the range -7FFFFFFFFF(16) through 7FFFFFFFFF(16), or

A hexadecimal string that contains a maximum of 12 hex digits (blanks do not count); each hex digit corresponds to four bits.

A hexadecimal integer must begin with a 0 digit and end with the radix specifier (16).

--ERROR DB 35-- An X register value must be 1..16 hex characters or 1..8 ASCII characters.

Condition Identifier: DB\$INVALID_XREG_VALUE Code: DB 35

Description: The value assigned to the VALUE parameter is too large.

User Action: Specify a value that is within the following ranges:

An integer in the range -7FFFFFFFFFFFFFFF(16) through 7FFFFFFFFFFFFFFF(16).

A hexadecimal string that contains a maximum of 16 hex digits (blanks do not count); each hex digit corresponds to four bits.

An ASCII string that contains a maximum of eight ASCII characters; each character corresponds to one byte.

A hexadecimal integer must begin with a 0 digit and end with the radix specifier (16).

--ERROR DB 39-- A write condition cannot be mixed with a non-read condition.

Condition Identifier: DB\$DEBUG_COND_CONFLICT Code: DB 39

Description: A conflict occurred with a non-read condition when you set a break on a write condition.

User Action: Delete the break set on the write condition.

--ERROR DB 40-- The byte offset is beyond the end of the procedure.

Condition Identifier: DB\$OFFSET_BEYOND_PROCEDURE Code: DB 40

Description: The base address plus the byte offset specifies an address that is beyond the end of the procedure.

User Action: Specify a smaller value for the byte offset.

--ERROR DB 41-- The specified stack frame was not found.

Condition Identifier: DB\$TARGET_SF_NUMBER_TOO_BIG Code: DB 41

Description: The value assigned to the START parameter is greater than the number of stack frames that can be displayed. The number of stack frames that can be displayed is the same as the number of calls in the active call chain.

User Action: Enter the DISPLAY_CALLS command to see how many calls are in the active call chain and then specify a value for the START parameter that is equal to or smaller than the number of calls.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 42-- Debug internal error - recursion levels exceeded.

Condition Identifier: DBE\$TRAPPED_INDEX_EXCEEDED Code: DB 42

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 43-- Debug internal error - redundant call to pop stack.

Condition Identifier: DBE\$REDUNDANT_CALL_TO_POP_STACK Code: DB 43

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 44-- The specified address is invalid or requested access to it is not allowed.

Condition Identifier: DBE\$INVALID_REQUEST Code: DB 44

Description: The specified address does not have the form of a valid address. Usually, an address is a hex integer that has the form: rsss0000000. r is the ring number, s is the segment, and 0000000 is the offset within the segment.

User Action: Specify a valid address.

Further Information: See the Debug Usage manual for information about the Debug command ADDRESS parameter.

--ERROR DB 45-- {text} is an invalid event identifier on a SET_BREAK command.

Condition Identifier: DBE\$UNIMPLEMENTED_EVENT_ID Code: DB 45

Description: Valid event identifiers are:

. arithmetic_overflow . arithmetic_significance . branch . call . divide_fault . execution . exponent_overflow . exponent_underflow . floating_point_indefinite . floating_point_significance . invalid_bdp_data . read . read_next_instruction . write

User Action: Specify a valid event identifier.

--ERROR DB 46-- {text}

Condition Identifier: DBE\$INVALID_PARAM_VALUE Code: DB 46

Description: The specified value is not valid for the corresponding parameter.

User Action: Review the requirements of the parameter value and then correct the value.

Further Information: For a description of the Debug utility, see the Debug Usage manual.

--ERROR DB 47-- {text} is not an address in the code section.

Condition Identifier: DBE\$NOT_A_CODE_ADDRESS Code: DB 47

Description: {Text} lies outside the address boundaries of the source program.

User Action: Check the source listing and then specify an address that lies within the address boundaries.

--CATASTROPHIC DB 48-- DEBUG ABORT.

Condition Identifier: DBE\$DEBUG_ABORT Code: DB 48

Description: A fatal error occurred. Debug could do nothing but abort. As much useful information as possible follows this message.

User Action: Follow the information provided. Report the problem if it reoccurs.

--ERROR DB 49-- The specified address is not in an active procedure.

Condition Identifier: DBE\$PVA_NOT_IN_KNOWN_PROC Code: DB 49

Description: The specified address does not match any address in the symbol table of an active procedure.

User Action: Specify the DISPLAY_CALLS command to find out which procedures are active and then specify an address that is in an active procedure.

Further Information: See the Debug Language Definition Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 51-- {text} is not an SCL variable name.

Condition Identifier: DBE\$SCL_VARIABLE_EXPECTED Code: DB 51

Description: {Text} does not conform to SCL variable name requirements. An SCL variable name can be a combination of from 1 to 31 letters, digits, or the special characters _ \$ # @ [] { } \ | ~ ^ It cannot begin with either a digit or a \$ character or be the names TRUE, FALSE, YES, NO, ON, or OFF.

User Action: Correct the name.

Further Information: See the NOS/VE System Usage manual for more information about SCL variable names.

--ERROR DB 52-- {text} is not an SCL string variable.

Condition Identifier: DBE\$SCL_STRING_VARIABLE_REQD Code: DB 52

Description: {Text} must be explicitly or implicitly created as an SCL string variable in the source program.

User Action: Use the SCL CREATE_VARIABLE command to explicitly create a string variable, or specify {text} as the object of an assignment statement, a control variable of a FOR statement, or a variable or array parameter to a procedure.

Further Information: See the NOS/VE System Usage manual for more information about explicitly or implicitly creating a string variable.

--ERROR DB 53-- Parameter value {text} exceeds 31 characters.

Condition Identifier: DBE\$PARAMETER_VALUE_TOO_LONG Code: DB 53

Description: The specified parameter value must be a valid SCL parameter value. An SCL parameter value cannot be longer than 31 characters.

User Action: Either correct the spelling of the parameter value or specify a parameter value that is shorter than 31 characters

--ERROR DB 54-- Identifier {text} is not active.

Condition Identifier: DBE\$VARIABLE_NOT_ACCESSIBLE Code: DB 54

Description: The specified identifier could not be located in a currently active procedure.

User Action: Check the source listing to see the correct module or procedure was specified. (If appropriate, enter the DISPLAY_DEBUGGING_ENVIRONMENT command to see the current default module and procedure.) Enter a DISPLAY_CALLS command to see if the identifier is in an active procedure.

--ERROR DB 55-- Identifier {text} was not found.

Condition Identifier: DBE\$VARIABLE_NOT_FOUND Code: DB 55

Description: The spelling of {text} does not match the spelling of any identifier in the source program.

User Action: Check the source listing for the correct spelling of the identifier.

--ERROR DB 56-- The Name parameter does not contain a variable identifier.

Condition Identifier: DBE\$EXPECTING_VARIABLE_NAME Code: DB 56

Description: The spelling of {text} does not match the spelling of any identifier in the source program.

User Action: Check the source listing for the correct spelling of {text}.

--ERROR DB 57-- {text} is not valid in an identifier.

Condition Identifier: DBE\$INVALID_TOKEN_IN_VARIABLE Code: DB 57

Description: {Text} is spelled with a character that is not valid in an identifier.

User Action: Check the source listing for the correct spelling of the identifier.

--ERROR DB 58-- Field specification is not allowed for {text}.

Condition Identifier: DBE\$ONLY_RECORDS_HAVE_FIELDS Code: DB 58

Description: {Text} cannot have a field specification because it was not defined as a record.

User Action: Remove the field specification.

--ERROR DB 59-- {text} is not a valid record field name.

Condition Identifier: DB\$NOT_VALID_FIELD_NAME Code: DB 59

Description: The spelling of {text} does not match the spelling of the record field name in the source program.

User Action: Check the source listing for the correct spelling of {text}.

--ERROR DB 60-- {text} is not a field of the current record.

Condition Identifier: DB\$FIELD_NOT_FOUND Code: DB 60

Description: The spelling of {text} does not match the spelling of the field name in the source program.

User Action: Check the source listing for the correct spelling of {text}.

--ERROR DB 61-- A pointer dereference is not allowed for {text}.

Condition Identifier: DB\$INVALID_POINTER_REFERENCE Code: DB 61

Description: {Text} is not a pointer and cannot be dereferenced.

User Action: Either use {text} without dereferencing it or correct the name of the pointer.

--ERROR DB 62-- A subscript is not allowed for {text}.

Condition Identifier: DB\$INVALID_SUBSCRIPT_REFERENCE Code: DB 62

Description: {Text} is not an array variable and, therefore, cannot have a subscript.

User Action: Either specify {text} without a subscript or check the source listing for the variable name that you intended to specify.

--ERROR DB 63-- The subscript for array {text} is not the right type.

Condition Identifier: DB\$INVALID_TYPE_FOR_SUBSCRIPT Code: DB 63

Description: The array subscript can be either a constant or a variable. The constant or variable cannot be of logical type.

User Action: Specify either a constant or a variable for the array subscript.

--ERROR DB 64-- The subscript is out of range for array {text}.

Condition Identifier: DB\$SUBSCRIPT_OUT_OF_RANGE Code: DB 64

Description: The subscript does not correspond to a value within array {text}.

User Action: Check the source listing for the bounds of array {text} and then specify a subscript that corresponds to a value within those bounds.

--ERROR DB 65-- Procedure {text} is not in module {text}.

Condition Identifier: DB\$PROC_NOT_IN_MODULE Code: DB 65

Description: The specified (or default) module does not contain the specified procedure in its symbol table.

User Action: Either check the spelling of the specified procedure or check that the specified procedure is the one you intended to specify.

--ERROR DB 66-- Procedure {text} is not active.

Condition Identifier: DB\$PROCEDURE_NOT_ACTIVE Code: DB 66

Description: The arguments of a procedure that is not in the active call chain cannot be displayed or changed.

User Action: Either specify the DISPLAY_CALLS command to see which procedures are active or wait until the procedure is active and then display or change the arguments.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 67-- {text}

Condition Identifier: DB\$DEBUG_INTERNAL_ERROR Code: DB 67

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 68-- {text} has an invalid pointer value.

Condition Identifier: DB\$INVALID_POINTER_VALUE Code: DB 68

Description: The pointer variable that was referenced contains an invalid value.

User Action: Display the pointer value with the DISPLAY_PROGRAM_VALUE command and then correct the pointer value.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_PROGRAM_VALUE command.

--ERROR DB 69-- The byte offset is beyond the end of the code section.

Condition Identifier: DB\$OFFSET_BEYOND_CODE_END Code: DB 69

Description: The base address plus the byte offset specifies an address that is beyond the end of the code section.

User Action: Specify a smaller value for the byte offset.

--ERROR DB 70-- PROCEDURE cannot be specified for bound modules without Debug tables.

Condition Identifier: DB\$PROCEDURE_SPECIFIED Code: DB 70

Description: The specified procedure name cannot be found by Debug because the modules have been bound.

User Action: Specify the bound module name instead of the specified procedure name.

Further Information: See the Debug Usage manual for information about addressing bound modules.

--ERROR DB 72-- The specified address is in the code or binding section.

Condition Identifier: DB\$PVA_IN_CODE_OR_BINDING_SEC Code: DB 72

Description: An address in the code or binding section cannot be accessed.

User Action: Check the load map for the correct address or enter the DISPLAY_PROGRAM_VALUE command specifying the variable name and the parameter TYPE=HEX. The starting address of the variable will be displayed along with the hexadecimal representation of the current contents of the variable.

--ERROR DB 73-- The PROCEDURE or LINE parameter must be specified for module {text}.

Condition Identifier: DB\$PROC_OR_LINE_SPECIFICATION Code: DB 73

Description: The PROCEDURE or LINE parameter must be combined with the MODULE parameter on a SET_BREAK command to uniquely identify the module.

User Action: Refer to the source listing and then specify the procedure or line number using the PROCEDURE or LINE parameter.

Further Information: See the Debug Usage manual for information about the Debug SET_BREAK command PROCEDURE or LINE parameter.

--ERROR DB 74-- The VALUE parameter contains an invalid hex character.

Condition Identifier: DB\$INVALID_CHAR Code: DB 74

Description: Valid hex characters are the digits 0 through 9, the characters A through F, and blanks.

User Action: Correct the invalid hex character.

--ERROR DB 76-- The {text} section cannot be found because module {text} is part of bound module {text}.

Condition Identifier: DB\$SECTION_REFERENCE Code: DB 76

Description: Section addressing cannot be used when modules are bound.

User Action: Use machine-level addressing.

Further Information: See the Debug Usage manual for information about addressing bound modules.

--ERROR DB 77-- The specified address is beyond the end of the segment.

Condition Identifier: DBE\$BYTE_OFFSET_VALUE_TOO_BIG Code: DB 77

Description: The base address plus the byte offset specifies an address that is larger than the block size.

User Action: Specify a smaller value for the byte offset.

--ERROR DB 78-- Address range of the specified break overlaps with range of break {text} for at least one event.

Condition Identifier: DBE\$OVERLAPPING_ADDRESS_RANGE Code: DB 78

Description: Overlapping breaks are not allowed.

User Action: Either correct the address range of the specified break or delete the break.

--ERROR DB 79-- Unknown break_name {text} cannot be deleted.

Condition Identifier: DBE\$UNKNOWN_BREAK Code: DB 79

Description: The specified break name does not match any of the break names currently known to Debug.

User Action: Enter the DISPLAY_BREAK command to see a list of the break names currently known to Debug and then specify the correct break name on the command.

--ERROR DB 80-- The specified value type cannot be {text}.

Condition Identifier: DBE\$VALUE_TYPE_CONFLICT Code: DB 80

Description: A valid value type is ASCII (A), HEX (H), or INTEGER (I).

User Action: Specify one of the valid value types.

--ERROR DB 81-- Debug input file {text} is empty or positioned at end_of_information: task terminated.

Condition Identifier: DBE\$TERMINATE_EOI_ON_INPUT Code: DB 81

Description: Debug cannot operate on an empty file. Also, files are not repositioned to the beginning-of-information after the first access of the file by Debug or after a Debug session.

User Action: Check the contents of the file or rewind the file.

--ERROR DB 82-- No traceback is possible - there are no active procedures.

Condition Identifier: DBE\$TRACEBACK_NOT_POSSIBLE Code: DB 82

Description: A DISPLAY_CALL command was issued after the program terminated. Most likely, a RUN command was in effect and the program completed execution, in which case control reverted to the operating system.

User Action: Do not issue the DISPLAY_CALL command.

--ERROR DB 83-- Variable {text} contains an invalid boolean value.

Condition Identifier: DBE\$INVALID_BOOLEAN_VALUE Code: DB 83

Description: Valid boolean values are the words TRUE and FALSE.

User Action: Specify the DISPLAY_MEMORY command to see the value of the variable and then correct the value.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_MEMORY command.

--ERROR DB 84-- Variable {text} contains an invalid ordinal value.

Condition Identifier: DBE\$INVALID_ORDINAL_VALUE Code: DB 84

Description: The ordinal value specified in {text} lies outside the range defined for variable {text}.

User Action: Change the ordinal value so that it lies within the bounds defined for variable {text}.

--ERROR DB 85-- Variable {text} is an unaligned real number.

Condition Identifier: DBE\$UNALIGNED_REAL Code: DB 85

Description: Real variables that are used in the \$PROGRAM_VALUE function must be byte-aligned.

User Action: Specify the DISPLAY_PROGRAM_VALUE command to see the real variable and then specify the variable with a matching byte-alignment.

--ERROR DB 86-- Variable {text} is an unaligned string.

Condition Identifier: DBE\$UNALIGNED_STRING Code: DB 86

Description: String variables that are used in either the CHANGE_PROGRAM_VALUE command or the \$PROGRAM_VALUE function must be byte-aligned.

User Action: Specify the DISPLAY_PROGRAM_VALUE command to see the string variable and then specify the variable with a matching byte-alignment.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_PROGRAM_VALUE command.

--ERROR DB 87-- Variable {text} is an unaligned pointer.

Condition Identifier: DBE\$UNALIGNED_POINTER Code: DB 87

Description: Pointer variables that are used in the \$PROGRAM_VALUE function must be byte-aligned.

User Action: Specify the DISPLAY_PROGRAM_VALUE command to see the pointer variable and then specify the variable with a matching byte-alignment.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_PROGRAM_VALUE command.

--ERROR DB 88-- Variable {text} is not a basic type and cannot be returned as a function value.

Condition Identifier: DBE\$INVALID_DATA_TYPE_FOR_FUNC Code: DB 88

Description: The value cannot be returned in the variable because it was not declared with the correct type.

User Action: Specify a basic data type when using the \$PROGRAM_VALUE function.

Further Information: See the appropriate Language Definition Usage manual for information about basic data types.

--ERROR DB 89-- The new value is not the right type for variable {text}.

Condition Identifier: DBE\$INVALID_TYPE_FOR_NEW_VALUE Code: DB 89

Description: The type of the VALUE parameter value must be the same as the type of the NAME parameter variable.

User Action: Specify a VALUE parameter value that is the same type as the NAME parameter variable.

--ERROR DB 90-- The variable {text} is not byte aligned.

Condition Identifier: DBE\$VALUE_NOT_ALIGNED Code: DB 90

Description: A variable that is not byte-aligned cannot be changed.

User Action: Specify the DISPLAY_PROGRAM_VALUE command to see the variable and then specify the variable with a matching byte-alignment.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_PROGRAM_VALUE command.

--ERROR DB 91-- The new value is out of range for variable {text}.

Condition Identifier: DBE\$VALUE_OUT_OF_RANGE Code: DB 91

Description: The value of the VALUE parameter value is too big to be assigned to the NAME parameter variable.

User Action: Specify a VALUE parameter value that is within the range of the NAME parameter variable.

--ERROR DB 92-- {text} is a constant and cannot be changed.

Condition Identifier: DBE\$CANNOT_CHANGE_CONSTANT Code: DB 92

Description: The program defines the specified name as a constant, not as a variable.

User Action: Check that the correct name was specified.

--ERROR DB 93-- {text} is a packed structure that cannot be changed.

Condition Identifier: DBE\$ONLY_CHANGE_PACKED_ELEMENTS Code: DB 93

Description: Packed structures cannot be changed with the CHANGE_PROGRAM_VALUE command, only packed elements.

User Action: Either change each element of the packed structure individually or use the CHANGE_MEMORY command to change the packed structure value.

Further Information: See the Debug Usage manual for information about both the Debug CHANGE_MEMORY command and packed structures.

--ERROR DB 94-- The value entered for the NAME parameter exceeded the allowed maximum of {text}.

Condition Identifier: DBE\$NAME_PARAMETER_TOO_LONG Code: DB 94

Description: The specified name is too long.

User Action: Check that the name is correct. To specify a very long name, assign the name to an SCL string variable and specify the string variable name prefixed by a ? character as the parameter value.

--ERROR DB 95-- The NAME parameter is required.

Condition Identifier: DBE\$NAME_PARAMETER_MISSING Code: DB 95

Description: The NAME parameter must be specified to uniquely identify the variable. The form of the NAME parameter is either NAME=variable or N=variable.

User Action: Specify the NAME parameter on the Debug command.

--ERROR DB 96-- An invalid NAME parameter was specified.

Condition Identifier: DBE\$INVALID_NAME_PARAMETER Code: DB 96

Description: The value specified for the NAME parameter is not valid.

User Action: Specify a valid NAME parameter value.

Further Information: See the Debug Usage manual for information about either the Debug CHANGE_PROGRAM_VALUE or DISPLAY_PROGRAM_VALUE command NAME parameter.

--ERROR DB 97-- An incorrect Cobol NAME Syntax was specified.

Condition Identifier: DBE\$BAD_VARIABLE_NAME_SYNTAX Code: DB 97

Description: The data-name does not conform to COBOL syntax.

User Action: Use the correct data-name format.

Further Information: See the COBOL Usage manual under COBOL Program Conventions for information about data-name syntax.

--ERROR DB 98-- {text} is not displayable. The variable was not passed to this entry point.

Condition Identifier: DBE\$VAR_NOT_PASSED Code: DB 98

Description: An actual parameter that was not passed to the subprogram cannot be displayed.

User Action: Do not attempt to display the variable.

--ERROR DB 99-- {text} is stored in a register and cannot be displayed.

Condition Identifier: DBE\$VAR_IN_REGISTER Code: DB 99

Description: A value that is not in memory cannot be displayed.

User Action: Use the DISPLAY_REGISTER command to find and display the value of {text}.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_REGISTER command.

--ERROR DB 100-- The pointer to the stack is not available, so {text} cannot be displayed.

Condition Identifier: DBE\$A4_NOT_SAVED Code: DB 100

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 101-- This set of qualifiers is not correct for {text}.

Condition Identifier: DBE\$BAD_QUALIFICATION_STRUCTURE Code: DB 101

Description: The data-name is not qualified correctly.

User Action: Qualify the data-name correctly.

Further Information: See the COBOL Usage manual under Qualification of User-Defined Words for information about data-name qualification.

--ERROR DB 102-- {text} is not uniquely qualified.

Condition Identifier: DBE\$QUALIFIER_NEEDED Code: DB 102

Description: A phrase must be added to uniquely qualify {text}. Examples of phrases are the IN data-name-2 and OF file-name phrases.

User Action: Specify {text} with qualifiers so that it is uniquely qualified.

Further Information: See the COBOL Usage manual under Qualification of User-Defined Words for information about data-name qualification.

--ERROR DB 103-- {text} must be numeric and non-indexed.

Condition Identifier: DBE\$NUMERIC_VARIABLE_REQUIRED Code: DB 103

Description: A data-name used as a subscript must be numeric and cannot be subscripted; a subscript cannot be subscripted.

User Action: Use the DISPLAY_PROGRAM_VALUE command to determine the value of the subscript, then specify the actual value for the subscript.

Further Information: See the Debug Usage manual for information about the DISPLAY_PROGRAM_VALUE command.

--ERROR DB 104-- A non-numeric value caused MATHLIB to return a {text} status.

Condition Identifier: DBE\$BDP_CONVERSION_ERROR Code: DB 104

Description: Nonnumeric data appears in a numeric data item. This can occur when a numeric item redefines a nonnumeric item or when the data-name has not been initialized.

User Action: Use the CHANGE_PROGRAM_VALUE command to replace the nonnumeric data with numeric data.

Further Information: See the Debug Usage manual for information about the Debug CHANGE_PROGRAM_VALUE command.

--ERROR DB 105-- The section is not in the module table.

Condition Identifier: DBE\$SECTION_ORDINAL_NOT_FOUND Code: DB 105

Description: This is a Debug internal error.

User Action: Report this error to your system analyst. The data-name that caused this error cannot be changed or displayed.

--ERROR DB 106-- {text} is not a static or parameter variable.

Condition Identifier: DBE\$UNIMPLEMENTED_VARIABLE_BASE Code: DB 106

Description: This is a Debug internal error.

User Action: Report this error to your system analyst. The data-name that caused this error cannot be changed or displayed.

--ERROR DB 107-- The wrong number of subscripts was specified for {text}.

Condition Identifier: DBE\$WRONG_NUMBER_OF_SUBSCRIPTS Code: DB 107

Description: Too many subscripts were specified for the data-name.

User Action: Specify the correct number of subscripts. Use the source listing to determine the correct number of subscripts.

--ERROR DB 108-- The specified subscript value is too big or too small.

Condition Identifier: DBE\$SUBSCRIPT_RANGE_ERROR Code: DB 108

Description: The subscript value is not within the allowed range.

User Action: Specify a subscript that is within the allowed range. Use the source listing to determine the allowed range.

--ERROR DB 109-- {text} is not referenced in the USING statement for a sub-program.

Condition Identifier: DBE\$VARIABLE_NOT_IN_USING_LIST Code: DB 109

Description: {Text} is in the Linkage Section of your subprogram, but is not in the USING phrase of the subprogram's Procedure Division header.

User Action: Recompile the subprogram with the data-name in the USING phrase or, if the data-name is in another program unit, use the MODULE parameter in the Debug command.

Further Information: See the COBOL Usage manual under Inter-Program Communication for information about subprogram communication.

--ERROR DB 110-- {text} reference modifier exceeds the scope of the variable.

Condition Identifier: DBE\$REFERENCE_MODIFIER_ERROR Code: DB 110

Description: The sum of the leftmost character position and the length, minus one of the reference modifier is greater than the number of characters in the data item.

User Action: Use the correct reference modifier on the command. Use the source listing to determine the length of the data item.

Further Information: See the COBOL Usage manual for information about reference modification.

--ERROR DB 111-- {text} cannot be retrieved from the stack because it was not saved.

Condition Identifier: DBE\$REGISTER_NOT_SAVED Code: DB 111

Description: Debug cannot determine which of the multiple entry points in a subprogram was actually called because the registers used to make the call were not saved. The arguments to the subprogram cannot be displayed or changed.

User Action: Display or change the variables in the calling program.

Further Information: See the Debug Usage manual for information about either the Debug DISPLAY_PROGRAM_VALUE or CHANGE_PROGRAM_VALUE command.

--ERROR DB 112-- DEBUG cannot find the ENTRY used for this subprogram.

Condition Identifier: DBE\$CANT_FIND_ENTRY Code: DB 112

Description: This is a Debug internal error. Debug tried to follow the path that was executed through the ENTRY statements to the subprogram where you are now trying to change a formal parameter.

User Action: Report this error to your system analyst.

--ERROR DB 113-- The substring reference is incorrectly formatted.

Condition Identifier: DBE\$INVALID_SUBSTRING Code: DB 113

Description: The substring specified in the NAME or VALUE parameter is not valid.

User Action: Correct the NAME or VALUE parameter substring.

Further Information: See the FORTRAN Language Definition Usage manual for information about substrings.

--ERROR DB 114-- The subscript reference is incorrectly formatted.

Condition Identifier: DBE\$INVALID_SUBSCRIPT Code: DB 114

Description: The subscript specified in the NAME or VALUE parameter value is not valid.

User Action: Correct the NAME or VALUE parameter subscript.

--ERROR DB 115-- Invalid data followed the NAME parameter.

Condition Identifier: DBE\$GARBAGE_AFTER_NAME Code: DB 115

Description: The NAME parameter variable has meaningless characters after it.

User Action: Remove the meaningless characters.

--ERROR DB 116-- An invalid complex number was found.

Condition Identifier: DBE\$INVALID_COMPLEX Code: DB 116

Description: A valid FORTRAN complex number is written as a pair of real or integer constants or symbolic constants separated by a comma and enclosed in parentheses, as follows:

(real,imag)

real is a real or integer constant or symbolic constant that represents the real part.

imag is a real or integer constant or symbolic constant that represents the imaginary part.

A plus or minus sign can precede either part.

User Action: Specify a valid complex number.

Further Information: See the FORTRAN Language Definition Usage Manual for information about complex constants.

--ERROR DB 117-- A bad status was returned by MATHLIB on an output conversion.

Condition Identifier: DBE\$BAD_MATHLIB_STATUS Code: DB 117

Description: This is a Debug internal error.

User Action: Report this error to your system analyst. Use the DISPLAY_MEMORY command to see the internal memory representation of the variable you want to convert.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_MEMORY command.

--ERROR DB 118-- No memory was allocated for this variable.

Condition Identifier: DBE\$NO_STORAGE Code: DB 118

Description: The compiler did not allocate space for the variable, therefore, the variable cannot be changed or displayed.

User Action: Check the reference map produced during compilation to see if the specified variable was listed as unused.

Further Information: See the FORTRAN Language Definition Usage manual for information about reference maps.

--ERROR DB 119-- Bit aligned values are not candidates for changing.

Condition Identifier: DBE\$BIT_ALLIGNED_ILLEGAL Code: DB 119

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 120-- It is illegal to change a variable of this type.

Condition Identifier: DBE\$CANT_CHANGE_TYPE Code: DB 120

Description: Variables that cannot be changed include constants, labels, subprogram names, and NAMELIST group names.

User Action: Check that the correct name was specified. You can use the DISPLAY_PROGRAM_VALUE command to see the contents of the variable, but you cannot change its contents.

Further Information: See the Debug Usage manual for information about the Debug CHANGE_PROGRAM_VALUE command.

--ERROR DB 121-- Arrays are not stored in the same order and cannot be changed.

Condition Identifier: DBE\$UNLIKE_STORAGE Code: DB 121

Description: Either the source array is stored row-wise and the destination array is shaped column-wise, or the source array is stored column-wise and the destination array is shaped row-wise.

User Action: Change the shape of the destination array to match the shape of the source array and then recompile the source program.

--ERROR DB 122-- The named array cannot be changed to one of unequal size.

Condition Identifier: DBE\$UNEQUAL_NUMBER_ELEMENTS Code: DB 122

Description: The array specified by the VALUE parameter must match in number and type of elements with the array specified by the NAME parameter.

User Action: Change the VALUE parameter specification to match the NAME parameter specification in number and type of array elements.

--ERROR DB 123-- Line {text} contains no code. The nearest line to set a break is {text}{text}{text}.

Condition Identifier: DBE\$LINE_EXTENT_ZERO Code: DB 123

Description: Code was not generated for the line you set a break on.

User Action: Set the break on one of the suggested lines.

--ERROR DB 124-- {text} does not support recursion.

Condition Identifier: DBE\$RECURSION_NOT_SUPPORTED Code: DB 124

Description: COBOL does not support recursion; the RECURSION_LEVEL and RECURSION_DIRECTION parameters cannot be used. If the final parameter is to be specified, specify the parameter using its parameter name, for example, STATUS=status_variable.

User Action: Remove any RECURSION_LEVEL or RECURSION_DIRECTION parameters from the command. If the STATUS parameter is to be specified, use the parameter name.

--ERROR DB 125-- Alphanumeric values must be enclosed by apostrophes.

Condition Identifier: DBE\$VALUE_NO_QUOTE Code: DB 125

Description: Alphanumeric values cannot be delimited by quote marks (") in Debug commands.

User Action: Use apostrophes (') to delimit the alphanumeric value.

--ERROR DB 126-- A reference modifier of length zero was detected in {text}.

Condition Identifier: DBE\$REFERENCE_MODIFIER_LENGTH_0 Code: DB 126

Description: The data-name used to specify the length in a reference modifier cannot be zero.

User Action: Use a non-zero value for length.

Further Information: See the COBOL Usage manual for information about reference modification.

--ERROR DB 127-- Both MODULE and PROCEDURE parameters may not be specified.

Condition Identifier: DBE\$MODULE_PARMS_CONFLICT Code: DB 127

Description: Both the MODULE and PROCEDURE parameters cannot be specified on a SET_BREAK command.

User Action: Specify only one of the MODULE or PROCEDURE parameters.

--ERROR DB 128-- The value entered for UNIT is incorrect.

Condition Identifier: DBE\$INVALID_UNIT Code: DB 128

Description: The UNIT parameter value must be either the keyword PROCEDURE or LINE. If the UNIT parameter is not specified, LINE is assumed.

User Action: Specify the keyword PROCEDURE or LINE as the UNIT parameter value.

--ERROR DB 129-- An internal error has occurred and step_mode cannot be turned off.

Condition Identifier: DBE\$STEP_MODE_NOT_TURNED_OFF Code: DB 129

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 130-- Module {text} does not exist.

Condition Identifier: DBE\$INVALID_MODULE Code: DB 130

Description: {Text} does not meet the requirements of a MODULE parameter variable.

User Action: Review the requirements of a MODULE parameter variable and then correct the variable.

Further Information: See the Debug Usage manual for information about the Debug command MODULE Parameter.

--ERROR DB 131-- Procedure {text} does not exist.

Condition Identifier: DBE\$INVALID_PROCEDURE Code: DB 131

Description: {Text} does not meet the requirements of a PROCEDURE parameter variable.

User Action: Review the requirements of a PROCEDURE parameter variable and then correct the variable.

Further Information: See the Debug Usage manual for information about the Debug command PROCEDURE parameter.

--ERROR DB 132-- Step mode is not available at this optimization level.

Condition Identifier: DBE\$NO_STYLIZED_CODE Code: DB 132

Description: Step mode cannot be used when the program was compiled with high optimization.

User Action: Re-compile the source program, specifying OPTIMIZATION_LEVEL=DEBUG.

--ERROR DB 133-- {text} execution, rni, call, branch, read, or write breaks must be deleted before step_mode is turned on.

Condition Identifier: DBE\$NOT_ENOUGH_HW_BREAKS Code: DB 133

Description: Step mode cannot be used because the maximum number of breaks that require the CYBER 180 debugging hardware have been set.

User Action: Delete the specified number of execution, read_next_instruction (rni), call, branch, read, or write breaks before you turn step mode on.

Further Information: See the Debug Usage manual for information about the Debug DELETE_BREAK command.

--ERROR DB 134-- MODULE or PROCEDURE is required to narrow the scope of step_mode to a manageable size.

Condition Identifier: DBE\$TOO_MANY_SECTIONS Code: DB 134

Description: Step mode cannot be activated because the number of modules in your source program would require Debug to set more than the maximum number of internal breaks allowed.

User Action: Specify the CHANGE_DEFAULTS command to change the default module then specify the SET_STEP_MODE command with MODULE=\$CURRENT.

Further Information: See the Debug Usage manual for information about the Debug CHANGE_DEFAULTS command.

--ERROR DB 135-- {text} breaks must be deleted before step_mode can be turned on.

Condition Identifier: DBE\$NOT_ENOUGH_BREAKS Code: DB 135

Description: The specified breaks must be deleted before step mode can be turned on so that overlapping breaks will not occur.

User Action: Delete the specified breaks and then re-enter the command.

--ERROR DB 136-- This break cannot be set. Procedure {text} as specified in module {text} contains no code.

Condition Identifier: DBE\$NO_CODE_IN_PROC Code: DB 136

Description: Procedure {text} does not contain any executable code; a break must be set on an executable line.

User Action: Do not set a break in procedure {text}.

--ERROR DB 137-- A list of values is not allowed for the VALUE parameter in this context.

Condition Identifier: DBE\$LIST_NOT_ALLOWED Code: DB 137

Description: The NAME variable is not of complex type; therefore, a list of values (which indicates a variable of complex type) is not valid as a VALUE variable.

User Action: Check to see that the correct NAME variable was specified or remove the second value from the VALUE variable.

--ERROR DB 138-- The value entered for the VALUE parameter is too long.

Condition Identifier: DBE\$VALUE_PARM_TOO_LONG Code: DB 138

Description: The specified value is too long for the specified variable.

User Action: Check that the correct program variable is specified. Specify a value for the variable that will fit into the variable as it is declared in the program.

--ERROR DB 139-- The exact size of array {text} is unknown at this time.

Condition Identifier: DBE\$ADJUST_ARRAY_SIZE_UNKNOWN Code: DB 139

Description: When a subprogram that contains an adjustable array is entered, the exact size of the array is unknown, and no array elements can be displayed or changed.

User Action: Display or change the array in the calling program by specifying the calling program name as the MODULE parameter variable and the array name as the NAME parameter variable. Or, display or change the adjustable array after the first line of the subprogram has executed.

--ERROR DB 140-- Assumed size arrays cannot be changed.

Condition Identifier: DBE\$CANT_CHANGE_ASSUMED_ARRAY Code: DB 140

Description: The values of the assumed size array elements cannot be changed because the size of the assumed size array is not known.

User Action: Change the assumed size array denotation with an array size and then re-compile the source program.

--ERROR DB 141-- The specified range for the NUMBER parameter is invalid.

Condition Identifier: DBE\$INVALID_RANGE_VALUE Code: DB 141

Description: A valid range of register numbers contains integers in ascending order.

User Action: Specify a valid range of registers with integers in ascending order. For example, NUMBER=(2..5).

--INFORMATIVE DB 142-- \$ALL is not a valid parameter value when module generator is {text}.

Condition Identifier: DBE\$ALL_NOT_IMPLEMENTED Code: DB 142

Description: \$ALL is not a valid parameter value for this particular language.

User Action: Remove the \$ALL parameter.

--ERROR DB 143-- No symbol table is available for the module.

Condition Identifier: DBE\$SYMBOL_TABLE_NOT_AVAILABLE Code: DB 143

Description: The source program was compiled without DEBUG_AIDS=DT. Symbol tables were not generated.

User Action: Recompile program specifying the DT option on the DEBUG_AIDS parameter.

--INFORMATIVE DB 144-- Variable values are not available for this system call.

Condition Identifier: DBE\$SYMBOL_TABLE_DOES_NOT_EXIST Code: DB 144

Description: The source program was compiled without DEBUG_AIDS=DT. Symbol tables were not generated; therefore, symbolic names cannot be specified.

User Action: Recompile program specifying the DEBUG_AIDS=DT parameter or use module or machine-level addressing.

--INFORMATIVE DB 145-- There are no {text} on the stack frame.

Condition Identifier: DBE\$CALL_NOT_ON_SF Code: DB 145

Description: No calls of the specified type currently on the stack frame.

User Action: Check that the correct call type was specified.

--INFORMATIVE DB 146-- The variable_values option is not implemented for this language.

Condition Identifier: DBE\$VV_NOT_IMPLEMENTED Code: DB 146

Description: The display_calls variable_values option is not available for the language of the program being debugged.

User Action: Not applicable.

--INFORMATIVE DB 147-- {text} not implemented.

Condition Identifier: DBE\$GENERIC_UNIMPLEMENTED Code: DB 147

Description: A feature has not been implemented.

User Action: None.

--ERROR DB 148-- Duplicate values specified for the {text} parameter.

Condition Identifier: DBE\$DUPLICATE_PARAMETER_VALUES Code: DB 148

Description: Only one value is allowed for this parameter type.

User Action: Remove one of the duplicate values.

--ERROR DB 149-- COBOL_PARAGRAPH or COBOL_SECTION feature only valid when debugging COBOL programs.

Condition Identifier: DBE\$NOT_A_COBOL_PROGRAM Code: DB 149

Description: The keywords COBOL_PARAGRAPH and COBOL_SECTION are valid only when debugging a COBOL program.

User Action: Remove all COBOL_PARAGRAPH and COBOL_SECTION keywords.

--ERROR DB 150-- \$ALL keyword cannot be used in a list.

Condition Identifier: DBE\$ALL_NOT_ALONE Code: DB 150

Description: The \$ALL keyword cannot be used in a list.

User Action: Remove \$ALL from list.

--ERROR DB 151-- The file {text} is empty.

Condition Identifier: DBE\$FS_SOURCE_FILE_EMPTY Code: DB 151

Description: The specified source file contains no program source text.

User Action: Check that the correct file was specified.

--ERROR DB 152-- {text} was given as the file processor. It is not a language DEBUG understands.

Condition Identifier: DBE\$FS_UNKNOWN_SOURCE_LANGUAGE Code: DB 152

Description: The name given for the file processor (the compiler for the language) is not one of those recognized by Full-Screen Debug.

User Action: Enter the name BASIC, C, COBOL, CYBIL, FORTRAN, or Pascal depending on the language to be used.

--ERROR DB 153-- The file designated as program source is not a source file.

Condition Identifier: DBE\$FS_UNKNOWN_SOURCE_FILE_TYPE Code: DB 153

Description: The specified file does not contain the source text of a program.

User Action: Check that the correct file was specified.

--ERROR DB 154-- Full-screen debugging is not available for {text} because it was compiled without debugging tables.

Condition Identifier: DBE\$FS_NO_TABLES Code: DB 154

Description: Full-screen message.

User Action: Recompile the program using the DEBUG_AIDS=DT option.

--ERROR DB 155-- Source file does not contain a valid COBOL program.

Condition Identifier: DBE\$FS_SOURCE_NOT_COBOL Code: DB 155

Description: The specified source file does not contain the text of a COBOL program.

User Action: Check that the correct file was specified.

--ERROR DB 157-- Statement labels of type {text} are not legal for this language.

Condition Identifier: DBE\$WRONG_LABEL_TYPE Code: DB 157

Description: Illegal statement label entered.

User Action: Verify that the specified statement label conforms to this language's legal statement types.

--ERROR DB 158-- Statement labels must be decimal integers.

Condition Identifier: DBE\$WRONG_RADIX_FOR_LABEL Code: DB 158

Description: Statement label that was not a decimal integer was entered.

User Action: Re-enter a valid statement label.

--ERROR DB 159-- Statement label {text} was not found.

Condition Identifier: DBE\$LABEL_NOT_FOUND Code: DB 159

Description: The specified statement label could not be located.

User Action: Inspect the program source to verify that the correct label was specified, and re-enter.

--ERROR DB 160-- There are no active procedures in module {text}. The PROCEDURE parameter is also needed.

Condition Identifier: DBE\$PROC_MUST_BE_SPECED Code: DB 160

Description: The command requires specification of the procedure (subprogram) in the specified module.

User Action: Specify the PROCEDURE parameter or remove the MODULE parameter so that the default module and procedure are used.

--ERROR DB 161-- {text} is not legal for the {text} parameter.

Condition Identifier: DBE\$ILLEGAL_PARAMETER_VALUE Code: DB 161

Description: Illegal value entered for a parameter.

User Action: Find the legal parameter values in the reference manual.

--ERROR DB 162-- Subscript syntax error was detected while scanning variable {text}.

Condition Identifier: DBE\$SUBSCRIPT_ERROR Code: DB 162

Description: The subscript specification does not follow the syntax rules for the programming language.

User Action: Correct the subscript.

--ERROR DB 163-- Unbalanced parenthesis in referencing {text}.

Condition Identifier: DBE\$UNBALANCED_PARENTHESIS Code: DB 163

Description: Every left parenthesis must have a matching right parenthesis in the reference.

User Action: Add a matching parenthesis or remove the unneeded parenthesis.

--ERROR DB 164-- The value of subscript number {text} of variable {text} is out of bounds.

Condition Identifier: DBE\$SUBSCRIPT_OUT_OF_BOUNDS Code: DB 164

Description: The subscript value is outside of the valid range of subscripts as declared for the program variable.

User Action: Correct the subscript value.

--ERROR DB 165-- The start of substring of {text} is incorrectly specified.

Condition Identifier: DBE\$INVALID_SUBSTRING_START Code: DB 165

Description: The substring start position is not in a valid range.

User Action: Re-enter the substring start position.

--ERROR DB 166-- The length of substring of {text} is incorrectly specified.

Condition Identifier: DBE\$INVALID_SUBSTRING_LENGTH Code: DB 166

Description: The specified substring length does not correspond to the defined string length.

User Action: Re-enter the substring length.

--ERROR DB 167-- The end of substring of {text} is incorrectly specified.

Condition Identifier: DBE\$INVALID_SUBSTRING_END Code: DB 167

Description: The substring ending position is not in a valid range.

User Action: Re-enter the substring ending position.

--ERROR DB 168-- The value of subscript number {text} is incorrectly specified.

Condition Identifier: DBE\$INVALID_SUBSCRIPT_SPEC Code: DB 168

Description: A subscript value is incorrect.

User Action: Correct the subscript value.

--ERROR DB 169-- The parameter {text} is not legal for this language.

Condition Identifier: DBE\$INVALID_PARAMETER Code: DB 169

Description: A parameter which is not supported by this language was entered.

User Action: Verify that the specified parameter conforms to this language's legal parameter types.

--ERROR DB 170-- {text} was not found because the segment access file is not open.

Condition Identifier: DBE\$INVALID_SEGMENT_ACCESS Code: DB 170

Description: The segment access file is not opened.

User Action: Open the segment access file.

--ERROR DB 171-- The NAME parameter contains an invalid identifier.

Condition Identifier: DBE\$INVALID_NAME_IDENTIFIER Code: DB 171

Description: An invalid identifier was entered in the NAME parameter.

User Action: Correct the NAME parameter.

--ERROR DB 172-- Errors encountered in display_program_value list of names.

Condition Identifier: DBE\$ERRORS_IN_LIST_OF_NAMES Code: DB 172

Description: An error occurred in the list of names in the display_program_value command.

User Action: Correct the error in the list of names.

--ERROR DB 173-- The VALUE parameter contains the invalid value {text}.

Condition Identifier: DBE\$INVALID_VALUE_CHARACTER Code: DB 173

Description: An invalid value has been entered for the VALUE parameter.

User Action: Correct the VALUE parameter.

--ERROR DB 174-- {text} is out of range.

Condition Identifier: DBE\$OUT_OF_RANGE Code: DB 174

Description: The complex constant specified is out of range.

User Action: Check the complex constant.

--ERROR DB 175-- {text} is an out of range value for variable {text}.

Condition Identifier: DBE\$OUT_OF_RANGE_NEW_VALUE Code: DB 175

Description: The VALUE parameter specification on the CHANGE_PROGRAM_VALUE command is too large.

User Action: Correct the VALUE parameter on the CHANGE_PROGRAM_VALUE command.

--ERROR DB 176-- The specified line does not contain statement {text}.

Condition Identifier: DBE\$STATEMENT_NUMBER_NOT_FOUND Code: DB 176

Description: The line specified by the LINE parameter on the SET_BREAK command does not contain the number of statements indicated by the STATEMENT parameter value.

User Action: Check that you are referencing the correct line in your program, or have specified the correct statement number.

Further Information: Refer to the description of the SET_BREAK command in the Debug usage manual.

--ERROR DB 177-- The statement parameter must be combined with the line parameter.

Condition Identifier: DBE\$STATEMENT_GOES_WITH_LINE Code: DB 177

Description: When the STATEMENT parameter is specified, the LINE parameter must also be specified since the STATEMENT parameter indicates a statement number on the line indicated by the LINE parameter.

User Action: Specify both parameters.

Further Information: Refer to the description of the SET_BREAK command in the Debug Usage manual.

--ERROR DB 178-- {text} is an invalid type for a complex number.

Condition Identifier: DBE\$INVALID_TYPE_FOR_COMPLEX Code: DB 178

Description: The value specified for the VALUE parameter on the CHANGE_PROGRAM_VALUE command is incorrect for a complex value. The format of a complex value is

(real-part,imaginary-part)

Each part is a real, integer, or symbolic constant.

User Action: Correct the VALUE parameter.

--ERROR DB 179-- {text} is not a string. The requested substring reference is illegal.

Condition Identifier: DBE\$SUBSTRING_ILLEGAL Code: DB 179

Description: The variable in the substring reference must be a variable, array, or array element of type character.

User Action: Correct the value used in the substring reference in your CYBIL program.

--ERROR DB 180-- The specified start of the substring for {text} is out of bounds.

Condition Identifier: DBE\$SUBSTRING_START_RANGE_ERR Code: DB 180

Description: The first character position specified is out of range for the character variable.

User Action: Correct the substring reference.

--ERROR DB 181-- The specified length of the substring for {text} is out of bounds.

Condition Identifier: DBE\$SUBSTRING_LENGTH_RANGE_ERR Code: DB 181

Description: The substring you specified is out of bounds of the character variable, array or array element.

User Action: Check the substring reference and the length of the character variable in the variable declaration.

Further Information: Refer to the description of substrings in the CYBIL usage manual.

--ERROR DB 182-- The substring start specifier for {text} must be a positive integer.

Condition Identifier: DBE\$SUBSTRING_START_IS_INT Code: DB 182

Description: The first character position in the substring reference must be a positive integer.

User Action: Check the substring reference in your CYBIL program.

--ERROR DB 183-- Variables must be from one to eight bytes long to be displayed as integers.

Condition Identifier: DBE\$CANT_DISPLAY_AS_INTEGER Code: DB 183

Description: The variable cannot be displayed in integer format because it is not 8 bytes long. Character, complex, double precision, and NAMELIST data cannot be displayed in integer format.

User Action: Do not specify TYPE=INTEGER for this variable.

--ERROR DB 184-- Variables must be eight bytes long to be displayed as reals.

Condition Identifier: DBE\$CANT_DISPLAY_AS_REAL Code: DB 184

Description: The variable cannot be displayed in real format because it is not 8 bytes long. Character, complex, and double precision data cannot be displayed in real format.

User Action: Do not specify TYPE=REAL for this variable on the DISPLAY_PROGRAM_VALUE command.

--ERROR DB 185-- Variable {text} can only be displayed as its natural type dictates.

Condition Identifier: DBESTYPE_CHANGE_ILLEGAL Code: DB 185

Description: The variable cannot be displayed in a type other than its type in the program. Additionally, character, complex, and double precision type data can not be displayed in integer or real format. NAMELIST data can not be displayed in integer format.

User Action: Change the TYPE parameter specification or omit the parameter. If the parameter is omitted, values will be displayed as they are typed in your program.

--ERROR DB 186-- HEX is the only legal value of the TYPE parameter in {text}.

Condition Identifier: DBESTYPE_EQUALS_HEX_ONLY Code: DB 186

Description: The only valid specification for the TYPE parameter is HEX in {text}.

User Action: The specification of a TYPE value other than HEX in either BASIC or COBOL programs is not supported. Correct the TYPE value as HEX for these languages.

--ERROR DB 187-- The & character must be first and only appear once.

Condition Identifier: DBE\$C_ILLEGAL_PTR_CONSTRUCTION Code: DB 187

Description: To reference the pointer to a variable rather than the variable value itself, you specify the variable name preceded by the address operator (&).

User Action: Put the & before the variable name with no intervening space. Remove any extra & characters.

Further Information: For further information, see the Debug description in the C/VE Reference Manual.

--ERROR DB 188-- The pointer {text} can only be modified once.

Condition Identifier: DBE\$C_MULTIPLE_PTR_MODS Code: DB 188

Description: A pointer reference can have only one value added to it. By adding the integer value n to the pointer, you reference the nth object of the array or the nth character in the string. (Objects and characters are counted from 0.)

User Action: Remove the extra modifications of the pointer.

Further Information: See the Debug description in the C/VE Usage manual.

--ERROR DB 189-- {text} is not a pointer. Pointer arithmetic is legal only for pointers and arrays.

Condition Identifier: DBE\$C_MUST_BE_POINTER Code: DB 189

Description: The expression attempted to add a value to a variable that is a pointer variable. An array name references a pointer pointing to the beginning of the array.

User Action: Check that the correct variable is specified. If the variable name is correct, remove the arithmetic operation from the reference.

Further Information: See the Debug description in the C/VE Usage manual.

--ERROR DB 190-- {text} is not an integer, the only valid type for arithmetic with pointers.

Condition Identifier: DBE\$C_WRONG_TYPE_FOR_PTR_MOD Code: DB 190

Description: Only integer variables or constants can be added to a pointer. When the integer n is added to a pointer, the expression references the nth object in the array or the nth character in the string. (Objects and characters are counted from 0.)

User Action: Correct the expression so that the value added to the pointer is an integer. An integer constant contains only digits and does not include a decimal point.

Further Information: For further information on specifying C program values, see the Debug description in the C/VE Usage manual.

--ERROR DB 191-- A pointer arithmetic overflow was detected.

Condition Identifier: DBE\$C_PTR_MOD_RANGE_ERR Code: DB 191

Description: The pointer arithmetic operation causes an arithmetic overflow condition. In a pointer arithmetic operation, an integer value n is multiplied by the size of the object to which the pointer points. The product is then added to the pointer value to produce a pointer to the nth object in the array.

User Action: Correct the integer value added to the pointer so that it does not reference an object beyond the end of the array or string. For example, for a three-object array, the integer should not be greater than 2 because objects are counted from 0.

Further Information: See the Debug description in the C/VE Usage manual.

--ERROR DB 192-- The empty expression () is not allowed.

Condition Identifier: DBE\$C_EMPTY_EXPRESSION Code: DB 192

Description: The command must specify a variable or pointer; it cannot specify only an empty expression consisting of a left and right parentheses.

User Action: Correct the command so that the variable or pointer to be changed is correctly specified.

Further Information: See the Debug description in the C/VE Usage manual.

--ERROR DB 193-- Unbalanced parentheses were detected.

Condition Identifier: DBE\$C_UNBALANCED_PARENS Code: DB 193

Description: The expression includes a left parenthesis without a matching right parenthesis or a right parenthesis without a matching left parenthesis.

User Action: Add a parenthesis to match the unbalanced parenthesis or remove the unbalanced parenthesis.

Further Information: See the Debug description in the C/VE Usage manual.

--ERROR DB 194-- Syntax error was detected in {text}.

Condition Identifier: DBE\$INCORRECT_VAR_REFERENCE Code: DB 194

Description: An error was detected in the syntax of the variable you specified.

User Action: Consult the format of the variable specifications in the BASIC usage manual.

--ERROR DB 195-- A substring of a null string has been specified.

Condition Identifier: DBE\$SUBSTRING_OF_NULL_STRING Code: DB 195

Description: A substring of a null string has been specified. A null string is a string constant or a string variable that has a length of zero.

User Action: Do not specify a substring of a null length string.

Further Information: Refer to the description of strings in the BASIC usage manual.

--INFORMATIVE DB 196-- Variable values are not available for this user call.

Condition Identifier: DBE\$SF_NOT_INITIALIZED Code: DB 196

Description: The source program was compiled without the DEBUG_AIDS=DT or DEBUG_AIDS=ALL option. Symbol tables were not generated; therefore, symbolic names cannot be displayed.

User Action: Recompile your program specifying DEBUG_AIDS=DT or DEBUG_AIDS=ALL or use module or machine-level addressing.

Further Information: Refer to the description of the compilation command in the appropriate usage manual.

--ERROR DB 197-- A SUBROUTINE or FUNCTION statement was missing in the source.

Condition Identifier: DBE\$FS_NO_SUBROUTINE_STMT Code: DB 197

Description: No SUBROUTINE or FUNCTION statement was found in the source program.

User Action: Add the SUBROUTINE statement (if the subprogram is a subroutine) or the FUNCTION statement (if the subprogram is a function) to your source program.

--ERROR DB 198-- The address operator is illegal for the expression "{text}".

Condition Identifier: DBE\$C_ILLEGAL_ADDRESS_OP Code: DB 198

Description: Debug does not allow use of the address operator (&) on an expression. The address operator must precede an unmodified variable name. For example, the reference &str is valid, but the reference &str+3 is invalid.

User Action: Remove the modification of the & object.

--ERROR DB 199-- Array {text} is not allocated.

Condition Identifier: DBE\$ARRAY_NOT_ALLOCATED Code: DB 199

Description: The array specified can not be referenced until it is allocated. An array is allocated after the array name appears in an ALLOCATE statement. The array becomes deallocated when the array name appears in a DEALLOCATE statement.

User Action: Since you can only reference the array after it is allocated, access the array between the ALLOCATE and DEALLOCATE statements. At this point you can reference the array elements.

Further Information: Refer to the description of allocatable arrays in the array chapter of the FORTRAN Version 2 usage manual.

--ERROR DB 200-- Array {text} is zero sized.

Condition Identifier: DBE\$ZERO_SIZED_ARRAY Code: DB 200

Description: The array specified is a zero-sized array and its elements can not be referenced.

User Action: Do not reference the elements of the array. If you did not intend for the array to be of size zero, check the dimension bounds on the array declaration in your program. If the upper bound of a dimension is less than the lower bound of the same dimension, the array is zero-sized.

Further Information: Refer to the description of array declarations in the FORTRAN Version 2 usage manual.

--ERROR DB 201-- The VALUE parameter may not be an array section.

Condition Identifier: DB\$ARRAY_SECTION_AS_VALUE Code: DB 201

Description: The VALUE parameter on the CHANGE_PROGRAM_VALUE command cannot specify an array section.

For the CHANGE_PROGRAM_VALUE command, the VALUE parameter can specify a variable or a constant. If the variable is a whole array, it can only be changed to another array of the same shape and type.

User Action: Do not use an array section reference as a VALUE parameter specification.

--ERROR DB 202-- An error was detected in section subscript number {text}.

Condition Identifier: DB\$INVALID_SECTION_SUBSCRIPT Code: DB 202

Description: The section selector in an array section reference is incorrect.

User Action: Correct the section selector. A section selector can be one of two forms:

- A one dimensional array of type integer.
- An indexed section selector of the form [s1:][s2]:[s3]

The section selector must not cause a reference to an element outside the bounds of the array being sectioned. If a section selector is a one dimensional array, all elements in the array must be within the bounds of the corresponding dimension of the array being sectioned.

For an indexed section selector, s1 must be greater than or equal to the lower bound of the corresponding dimension of the array being sectioned; s2 must be less than or equal to the upper bound of the corresponding dimension of the array being sectioned. The increment, s3, is optional.

Further Information: Refer to the description of array sections in the FORTRAN Version 2 usage manual.

--ERROR DB 203-- The increment of section subscript number {text} must be greater than zero.

Condition Identifier: DB\$BAD_SUBSCRIPT_INCREMENT Code: DB 203

Description: The increment specified in the indexed section selector must be a positive value.

User Action: Correct the increment value on the indexed section selector. An indexed section selector has the form:

[s1:][s2]:[s3]

The increment specification is optional; if you omit it, the default value is one.

Further Information: Refer to the description of array sections in the array chapter of the FORTRAN Version 2 usage manual.

--ERROR DB 204-- At least one value of section subscript number {text} is out of bounds.

Condition Identifier: DB\$BAD_SECTION_SUBSCRIPT_BOUND Code: DB 204

Description: The section subscript is causing a reference outside the bounds of the array being sectioned.

User Action: Correct the section subscript. A section subscript can be a section selector of one of two forms:

- A one dimensional array of type integer.
- An indexed section selector of the form [s1:][s2]:[s3]

The section selector must not cause a reference to an element outside the bounds of the array being sectioned. If a section subscript is a one dimensional integer array, all elements in the one dimensional integer array must be within the bound of the corresponding dimension of the array being sectioned.

For an indexed section selector, s1 must be greater than or equal to the lower bound of the corresponding dimension of the array being sectioned; s2 must be less than or equal to the upper bound of the corresponding dimension of the array being sectioned. The increment, s3, must not equal zero.

Further Information: Refer to the description of array sections in the array chapter of the FORTRAN Version 2 usage manual.

--ERROR DB 205-- In section subscript number {text}, the lower bound is incorrectly greater than the upper bound.

Condition Identifier: DBE\$LOWER_GREATER_THAN_UPPER Code: DB 205

Description: The section selector you have specified is incorrect; the lower bound is greater than the upper bound.

User Action: Correct the section selector. An indexed section selector is of the form

[s1:][s2]:[s3]

The increment specification is optional, if you omit it, the default value is one.

Further Information: Refer to the description of array sections in the array chapter of the FORTRAN Version 2 manual.

--ERROR DB 206-- The first character position must be positive in MID\$ reference.

Condition Identifier: DBE\$FIRST_POSITION_NOT_POSITIVE Code: DB 206

Description: The value you specified as the first character position in the MID\$ function reference must be a positive integer.

User Action: Specify a positive integer as the starting position in the MID\$ function reference.

Further Information: Refer to the description of MID\$ in the BASIC usage manual.

--ERROR DB 207-- The starting position must be within the string length.

Condition Identifier: DBE\$FIRST_POSITION_BEYOND_LEN Code: DB 207

Description: The first character position in the MID\$ substring reference is out of bounds of the host string.

User Action: Check the substring reference in the MID\$ specification.

Further Information: Refer to the description of substrings in the BASIC reference manual.

--ERROR DB 208-- The length in the MID\$ reference must not be negative.

Condition Identifier: DBE\$NEGATIVE_LENGTH Code: DB 208

Description: The value specified as the length in the MID\$ reference must be a positive integer or zero. If you specify zero as the length, a null string is returned.

User Action: Do not specify a negative value for length in the MID\$ function.

Further Information: Refer to the description of the MID\$ function in the BASIC usage manual.

--ERROR DB 209-- The REGISTER_NUMBER parameter is illegal for REGISTER_TYPE = P.

Condition Identifier: DBE\$ILLEGAL_REG_NUM_FOR_P Code: DB 209

Description: If P is specified on the \$REGISTER function for the type of register, a number must not be specified because there is only one P register.

User Action: Remove the number specified on the \$REGISTER function.

Further Information: See the Debug Usage manual for a description of the \$REGISTER function.

--ERROR DB 210-- Module {text} was not found in the DEBUG tables. The binary does not match the source file.

Condition Identifier: DBE\$FS_MATCHING_MOD_NOT_FOUND Code: DB 210

Description: You specified a source file which contained a different source program than the one being debugged (but written in the same language).

User Action: Specify the correct the source file.

--ERROR DB 211-- Function {text} is not active.

Condition Identifier: DBE\$FUNCTION_NOT_ACTIVE Code: DB 211

Description: A function has been referenced that is not active (is not currently being called).

User Action: Either specify the DISPLAY_CALLS command to see which functions are active or wait until the function is active and then display or change the arguments.

Further Information: See the Debug Usage manual for information about the Debug DISPLAY_CALLS command.

--ERROR DB 212-- The NAME parameter may not be an array section.

Condition Identifier: DBE\$ARRAY_SECTION_AS_NAME Code: DB 212

Description: The name of an array section cannot be specified for the NAME parameter of the CHANGE_PROGRAM_VALUE command.

User Action: Do not specify an array section name.

Further Information: Refer to the description of array sections in the FORTRAN Version 2 usage manual.

--ERROR DB 213-- The requested register is not in the stack.

Condition Identifier: DBE\$REG_NOT_IN_STACK Code: DB 213

Description: You cannot specify a register that is not saved on the \$REGISTER function.

User Action: Specify a saved register.

--ERROR DB 214-- For integers, the maximum number of bytes is eight.

Condition Identifier: DBE\$TOO_MANY_BYTES_FOR_INT Code: DB 214

Description: You cannot specify more than 8 bytes of memory to be displayed in integer format. The maximum integer size is 8 bytes.

User Action: Specify 8 or less bytes of memory to be displayed.

--ERROR DB 215-- Syntax error detected while scanning "{text}" for set element values.

Condition Identifier: DBE\$SET_VALUE_SYNTAX_ERR Code: DB 215

Description: A syntax error was found in the value specified in the CHANGE_PROGRAM_VALUE command of a set.

User Action: Check to see if an illegal character is specified for the set. For example, CHAPV set1 +[?]; the question marks are illegal.

--ERROR DB 216-- Set element {text} is the wrong type for the set variable {text}.

Condition Identifier: DBE\$SET_VALUE_TYPE_ERROR Code: DB 216

Description: The value specified in a CHANGE_PROGRAM_VALUE command of a set was not the same type as the set. For example, CHAPV booleanset value = +[9] changes a boolean set to an integer. This change is invalid.

User Action: Make sure the value specified is the same type as the set.

--ERROR DB 217-- {text} is not a legal set element value.

Condition Identifier: DBE\$ILLEGAL_SET_ELEMENT_VALUE Code: DB 217

Description: You cannot specify a string longer than 1 character for the set element.

User Action: Make sure the set element is only 1 character long.

--ERROR DB 218-- {text} is out of range for the set variable {text}.

Condition Identifier: DBE\$SET_VALUE_OUT_OF_RANGE Code: DB 218

Description: You cannot change a set using a set element which is not in the range of valid set elements.

User Action: Specify a set_element which is in the range of valid set elements.

--ERROR DB 219-- No user modules were found in the program.

Condition Identifier: DBE\$FS_NO_USER_MODULES Code: DB 219

Description: You cannot switch to user mode because no user modules exist.

User Action: None.

--ERROR DB 220-- {text} is not the processor of {text}.

Condition Identifier: DBE\$FS_FILE_PROC_MISMATCH Code: DB 220

Description: The FILE_PROCESSOR attribute of the file containing the program to be debugged must be the name of the compiler that compiled the program.

User Action: Change the FILE_PROCESSOR using the SCL CHANGE_FILE_ATTRIBUTES command.

Further Information: See the Debug Usage manual for FILE_PROCESSOR attribute requirements.

--ERROR DB 221-- {text} is already initialized.

Condition Identifier: DBE\$FS_MODULE_ALREADY_INIT Code: DB 221

Description: This is a Debug internal error.

User Action: Report this to your system analyst.

--ERROR DB 222-- A {text} statement was not found in the source.

Condition Identifier: DBE\$FS_MODULE_STMT_NOT_FND Code: DB 222

Description: You specified a file containing source code of a different language than the binary being executed.

User Action: Specify the correct file containing the correct source.

--ERROR DB 223-- A bad {text} statement was found in the source.

Condition Identifier: DBE\$FS_BAD_MODULE_STMT Code: DB 223

Description: The specified file does not contain the correct source to be debugged.

User Action: Check that the specified file contains the correct source.

--ERROR DB 224-- The module {text} in the source does not match the binary.

Condition Identifier: DBE\$FS_MODEND_NOT_FOUND Code: DB 224

Description: The source code in the file specified does not match the binary code being executed.

User Action: Make sure the correct file containing the correct source is specified.

--ERROR DB 225-- The file {text} does not exist.

Condition Identifier: DBE\$FS_SOURCE_FILE_NOT_FOUND Code: DB 225

Description: Full-screen message displayed in Output window when the file containing the source code to be debugged does not exist.

User Action: None.

--ERROR DB 226-- "{text}" is not a properly formed address.

Condition Identifier: DBE\$IMPROPER_ADDRESS Code: DB 226

Description: The input you specified on the address parameter is longer than the longest valid binary address.

User Action: Make sure the address specified is within the valid binary address range.

--ERROR DB 227-- No symbol was found whose address matches the pva given.

Condition Identifier: DBE\$NO_SYMBOL_FOUND Code: DB 227

Description: This is a Debug internal error.

User Action: Report this to your system analyst.

--ERROR DB 228-- The pva found could not be symbolically formatted.

Condition Identifier: DBE\$VARIABLE_UNFORMATTABLE Code: DB 228

Description: This is a Debug internal error.

User Action: Report this to your system analyst.

--ERROR DB 229-- A line on which to set a break when the specified procedure becomes active could not be found.

Condition Identifier: DBE\$CANT_FIND_LINE_FOR_PROC Code: DB 229

Description: Debug internal error while trying to set up a read/write break on a stack frame based variable.

User Action: Report to site analyst.

--ERROR DB 230-- {text} {text} call, branch, rni, read, or write breaks must be deleted before this break can be set.

Condition Identifier: DBE\$NEED_MORE_HW_BREAKS Code: DB 230

Description: The break you requested causes an overflow in the Debug break table.

User Action: Delete at least the event specified and reenter the command. Remember to delete unneeded breaks since each break set slows the execution of your program.

--ERROR DB 231-- {text} {text} breaks must be deleted before this one can be set.

Condition Identifier: DBE\$NEED_MORE_BREAKS Code: DB 231

Description: The break you requested causes an overflow in the Debug break table.

User Action: Delete at least the number specified and reenter the command. Remember to delete unneeded breaks since each break set slows the execution of your program.

--ERROR DB 232-- Variable references specified for breaks cannot contain non literal qualifiers.

Condition Identifier: DBE\$NON_LITERAL_QUAL_ILLEGAL Code: DB 232

Description: You cannot specify an array subscript or substring reference as a variable on the SET_BREAK command.

User Action: Substitute a constant or literal value for the subscript or substring reference.

Further Information: See the Debug Usage manual for a description of the SET_BREAK parameters.

--ERROR DB 233-- Array sections cannot be specified for the NAME parameter.

Condition Identifier: DBE\$NOT_ALLOWED_ON_ARRAY_SEC Code: DB 233

Description: The variable specified on the NAME parameter cannot be an array section. However, the variable name can be simple unsubscripted variable names, subscripted names (subscripts can be constants or variables, but not expressions), field references, and pointer references only.

User Action: Correct the command by replacing the constant specified on the NAME with a valid variable name.

Further Information: See the Debug Usage manual for the NAME parameter restrictions.

--ERROR DB 234-- A break cannot be set for a variable reference of this type.

Condition Identifier: DBE\$WRONG_TYPE_FOR_BREAK Code: DB 234

Description: You cannot set a READ or WRITE break on a:

- FORTRAN namelist, assumed-size array, assumed-shape array, or array section
- BASIC array or string
- Constant
- Variable with length of bits or length of zero
- Substring or subscripts that use non-literal qualifiers

User Action: Specify the correct variable type for the READ or WRITE break.

--ERROR DB 235-- A break cannot be set on an assumed size array.

Condition Identifier: DBE\$NOT_ON_ASSUMED_SIZE Code: DB 235

Description: A READ or WRITE break cannot be set on assumed size arrays.

User Action: Do not specify a READ or WRITE break on an assumed-size array.

Further Information: See the Debug usage manual for a description of the SET_BREAK command.

--ERROR DB 236-- The NAME parameter can only be combined with the MODULE parameter.

Condition Identifier: DBE\$NAME_GOES_WITH_MODULE Code: DB 236

Description: The MODULE parameter is specified to qualify the NAME parameter. Therefore, the MODULE parameter can be specified only if the NAME parameter is specified.

User Action: Make sure the NAME parameter is specified also.

--ERROR DB 237-- {text} is a bound module with no DEBUG tables and therefore cannot be debugged.

Condition Identifier: DBE\$FS_OBJ_LIB_NO_TABLES Code: DB 237

Description: You cannot select the Zoom-in display for a module that has not been compiled with Debug tables. For example, if you switch to system mode (selected in the OPTIONS display) and select the Zoom-in display, the system module may not include Debug tables.

User Action: Specify a bound module with Debug tables.

--ERROR DB 238-- SCOPE = {text} is not valid in this language.

Condition Identifier: DBE\$LANG_ILLEG_FOR_PAR_OF_SCOPE Code: DB 238

Description: The SCOPE=MODULE parameter option is not supported by the BASIC, C, COBOL, FORTRAN, and Pascal languages.

The SCOPE=LOCAL parameter option is not supported by the C and FORTRAN languages.

User Action: Correct the command by removing the SCOPE=MODULE or SCOPE=LOCAL parameter options.

Further Information: See the Debug Usage manual for the SCOPE=MODULE and SCOPE=LOCAL restrictions.

--ERROR DB 239-- SCOPE = MODULE may not be combined with the {text} parameter.

Condition Identifier: DBE\$ILLEGAL_COMB_WITH_MODULE Code: DB 239

Description: The SCOPE=MODULE parameter option cannot be used with the SECTION, PROCEDURE, RECURSION_LEVEL, and RECURSION_DIRECTION parameters or with the NAME_OPTION=PARAMETERS parameter option.

User Action: Correct the command by removing the SCOPE=MODULE parameter option.

Further Information: See the Debug Usage manual for the SCOPE=MODULE restrictions.

--ERROR DB 240-- SCOPE = GLOBAL may not be combined with the {text} parameter.

Condition Identifier: DBE\$ILLEGAL_COMB_WITH_GLOB Code: DB 240

Description: The parameter option SCOPE=GLOBAL cannot be used with the MODULE, PROCEDURE, RECURSION_LEVEL, and RECURSION_DIRECTION parameters or with the NAME_OPTION=PARAMETERS parameter option.

User Action: Correct the command by removing the SCOPE=GLOBAL parameter option.

Further Information: See the Debug Usage manual for a description of the SCOPE=GLOBAL restrictions.

--ERROR DB 241-- The SCOPE parameter is not valid for programs written in C.

Condition Identifier: DBE\$SCOPE_ILLEGAL_IN_C Code: DB 241

Description: The SCOPE parameter is not supported by the C language.

User Action: Correct the command by removing the SCOPE parameter.

Further Information: See the Debug Usage manual for the SCOPE parameter restrictions.

--ERROR DB 242-- SCOPE = {text} may not be used in combination with the NAME_OPTION value of PARAMETER.

Condition Identifier: DBE\$NAME_OPT_PARAM_ILLEGAL Code: DB 242

Description: The SCOPE=GLOBAL and SCOPE=MODULE parameter options cannot be used with the NAME_OPTION=PARAMETER parameter option.

User Action: Correct the command by removing the SCOPE=GLOBAL or SCOPE=MODULE parameter options.

Further Information: See the Debug Usage manual for the SCOPE=GLOBAL and SCOPE=MODULE restrictions.

--ERROR DB 243-- Breaks may not be set symbolically on variable references bit aligned or of length bits.

Condition Identifier: DBE\$BRK_NOT_ALLOWED_ON_BITS Code: DB 243

Description: A read/write break cannot be set on the specified variable because of its data structure.

User Action: None.

Further Information: See the Debug Usage manual for the SET_BREAK describing read and write breaks.

--ERROR DB 244-- Break {text} must be deleted before the requested break can be set.

Condition Identifier: DBE\$MUST_DELETE_THIS_BREAK Code: DB 244

Description: For breaks that do not have permanent memory addresses, Debug must set one or more internal breaks to insure that the break you request only accesses the name specified when it has a valid memory address. The internal break Debug must set is at the same location as the break referenced in the diagnostic.

User Action: Delete the break specified in the diagnostic and reenter the SET_BREAK command.

--ERROR DB 245-- The specified break could not be set for all entry points of a multiple entry point subprogram.

Condition Identifier: DBE\$NOT_AT_ALL_ENTRIES Code: DB 245

Description: You attempted to set a break using the name parameter in a FORTRAN subprogram with multiple entry points. Debug will not be able to honor the break at all the entries.

User Action: Continue execution realizing that the break will not always be honored or delete the break and use some other method to stop in the multiple entry point procedure and display program variable values.

--ERROR DB 246-- A break cannot be set on a variable reference whose length is zero.

Condition Identifier: DBE\$NOT_WHEN_LENGTH_ZERO Code: DB 246

Description: A READ or WRITE break cannot be set on variables of zero length.

User Action: Do not set a READ or WRITE break on a variable of zero-length

Further Information: See the Debug Usage manual for a description of the SET_BREAK command.

--ERROR DB 247-- Breaks are only allowed on non static variables in multiple entry point subprograms compiled at optimization_level=debug.

Condition Identifier: DB\$ONLY_WHEN_OPT_IS_DEBUG Code: DB 247

Description: You cannot set a READ or WRITE break on non-static variables in multiple entry point subroutines or functions if OPTIMIZATION_LEVEL=HIGH or OPTIMIZATION_LEVEL=LOW.

User Action: Compile the program at OPTIMIZATION_LEVEL=DEBUG.

--INFORMATIVE DB 248-- The specified break could not be set for the ENTRY statement at line {text}.

Condition Identifier: DB\$NOT_THIS_ENTRY Code: DB 248

Description: You attempted to set a break using the name parameter in a FORTRAN subprogram with multiple entry points. Debug will not be able to honor the break at the specified entry.

User Action: Continue execution realizing that the break will not always be honored or delete the break and use some other method to stop in the multiple entry point procedure and display program variable values.

--ERROR DB 249-- DISPV \$ALL SCOPE = GLOBAL is invalid.

Condition Identifier: DB\$SCOPE_EQ_GLOBAL_INVALID Code: DB 249

Description: You cannot specify SCOPE=GLOBAL when the NAME parameter value is \$ALL on the DISPLAY_PROGRAM_VALUE command.

User Action: Remove SCOPE=GLOBAL or specify another value for the NAME parameter.

Further Information: See the Debug Usage manual for a description of the DISPLAY_PROGRAM_VALUE command.

--WARNING DB 250-- Screen mode debugging could not be automatically initiated.

Condition Identifier: DB\$CANT_INIT_SCREEN Code: DB 250

Description: CHANGE_INTERACTION_STYLE is set to screen. Debug encountered an error trying to bring up screen mode automatically. Probably indicates a second task being initiated in screen mode.

User Action: Make sure Debug is the only task being initiated in screen mode.

--ERROR DB 252-- The SECTION parameter can only be combined with \$ALL.

Condition Identifier: DB\$SECTION_LEGAL_WITH_ALL Code: DB 252

Description: The SECTION parameter is valid only when the value of the NAME parameter is \$ALL.

User Action: Correct the command by removing the SECTION parameter or changing the value of the NAME parameter to \$ALL.

Further Information: See the Debug Usage manual for the SECTION parameter restrictions.

--ERROR DB 253-- {text} may not be specified with the SECTION parameter.

Condition Identifier: DB\$ILLEGAL_COMB_WITH_SECTION Code: DB 253

Description: The SECTION parameter cannot be used with the RECURSION_LEVEL and RECURSION_DIRECTION parameters or with the NAME_OPTION=PARAMETERS parameter option.

User Action: Correct the command by removing the SECTION parameter.

Further Information: See the Debug Usage manual for the SECTION parameter restrictions.

--ERROR DB 254-- SECTION may not be used in combination with the NAME_OPTION value of PARAMETER.

Condition Identifier: DB\$PARAM_ILLEGAL_WITH_SECTION Code: DB 254

Description: The SECTION parameter cannot be used with the NAME_OPTION=PARAMETERS parameter option.

User Action: Correct the command by removing the SECTION parameter.

Further Information: See the Debug Usage manual for the restrictions on the SECTION parameter.

--ERROR DB 255-- A constant may not be specified for the NAME parameter.

Condition Identifier: DBE\$CANT_SETB_ON_CONSTANT Code: DB 255

Description: The NAME parameter of the SET_BREAK command must specify a variable name only.

User Action: Correct the command so that the NAME parameter specifies a variable name.

Further Information: See the SET_BREAK command description in the Debug Usage manual.

--ERROR DB 256-- The specified entry point was not found.

Condition Identifier: DBE\$SPEC_ENTRY_PT_NOT_FOUND Code: DB 256

Description: The spelling of the specified entry point does not match the spelling of an entry point name in the source program.

The entry point may not exist if the module containing the entry point has not been loaded. Also, entry points can be deleted when a program is bound (see the CREATE_MODULE and CHANGE_MODULE_ATTRIBUTES subcommands of the CREATE_OBJECT_LIBRARY command).

User Action: Check the source listing for the spelling of the entry point name.

--ERROR DB 257-- The name_option CONSTANTS is not supported for {text} programs.

Condition Identifier: DBE\$ILLEGAL_NAME_OPTION Code: DB 257

Description: DISPLAY_PROGRAM_VALUE=\$ALL NAME_OPTION=CONSTANTS is not supported by the BASIC, C, and COBOL languages.

User Action: Do not specify NAME_OPTION=CONSTANTS for these languages.

--ERROR DB 258-- Identifier {text} was not found to match the specified name_options.

Condition Identifier: DBE\$VAR_DOES_NOT_QUALIFY Code: DB 258

Description: The NAME_OPTION value must match type with the NAME value.

User Action: Correct the command by specifying matching types for the NAME_OPTION and NAME parameter values.

Further Information: See the Debug Usage manual for a description of the DISPLAY_PROGRAM_VALUE command.

--ERROR DB 259-- In the current procedure error(s) were found setting break(s) {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text}

Condition Identifier: DBE\$ERRORS_SETTING_AUTO_BRKS Code: DB 259

Description: Debug internal error.

User Action: Report to your site analyst.

--ERROR DB 260-- There is no code associated with label {text}.

Condition Identifier: DBE\$NO_CODE_FOR_LABEL Code: DB 260

Description: The label chosen using the SET_BREAK command does not contain code.

User Action: Choose a different label or a line number to set the break on.

--ERROR DB 261-- {text} is defined in another module and its entry point does not exist. Use the MODULE parameter to specify the defining module.

Condition Identifier: DBE\$SEP_FOR_XREFED_VAR_NOT_FOUND Code: DB 261

Description: The specified variable defined in this module is external. External variables are generally found by using the entry point table, but in this case, the variable is not there. This may be because the module in which the variable is defined is not loaded or that the module was bound without retaining this entry point.

User Action: Re-enter the command using the MODULE parameter to specify the module where the variable is defined or re-run the program with the missing module loaded.

--INFORMATIVE DB 262-- {text} is defined in another module.

Condition Identifier: DBE\$SEP_FOR_XREFED_NOT_FND_DAN Code: DB 262

Description: The specified variables defined in this module are external. External variables are generally found by using the entry point table, but in this case, the variables are not there. This may be because the module in which the variables are defined is not loaded or that the module was bound without retaining this entry point.

User Action: Re-enter the command using the MODULE parameter to specify the module where the variables are defined or re-run the program with the missing module loaded.

--ERROR DB 263-- A call, branch, rni, read, or write break must be deleted before this break can be set.

Condition Identifier: DBE\$NEED_ANOTHER_HW_BREAK Code: DB 263

Description: There are 64 available breaks, 32 of which can be hardware breaks. In this case, either there are already 32 hardware breaks set, or the break you request requires "n" breaks and there are already "32-n+1" breaks set.

User Action: Delete a call, branch, rni, read, or write break before you set another break.

--ERROR DB 264-- A break must be deleted before this one can be set.

Condition Identifier: DBE\$NEED_ANOTHER_BREAK Code: DB 264

Description: There are 64 available breaks, 32 of which can be hardware breaks. Some breaks use more than one break. In this case, either there are already 32 break set or, the break you request requires "n" breaks and there are already "32-n+1" breaks set.

User Action: You must delete a break before you can set another break.

--ERROR DB 265-- {text} is not a legal command when debugging in full screen mode.

Condition Identifier: DBE\$ILLEGAL_FULL_SCREEN_CMD Code: DB 265

Description: The indicated command cannot be used while screen mode Debug is active.

User Action: Do not use the indicated command.

--ERROR DB 266-- There is no code associated with line number {text}.

Condition Identifier: DBE\$NO_CODE_FOR_LINE_NUMBER Code: DB 266

Description: The line number chosen using the SET_BREAK command does not contain code.

User Action: Choose a different line number to set the break on.

--ERROR DB 267-- Incorrect value(s) specified for the task number parameter: {text}.

Condition Identifier: DBE\$BAD_TN_PARAMETER Code: DB 267

Description: All the values entered for the TASK_NUMBER parameter of the DISPLAY_DEBUG_TASK_STATUS command are incorrect.

User Action: Use no TASK_NUMBER to see the current task or TASK_NUMBER=ALL to see all tasks.

--ERROR DB 269-- Screen mode debugging is not available in more than one task simultaneously.

Condition Identifier: DBE\$CANT_HAVE_SCREEN Code: DB 269

Description: Only one task of a multi-task application can be debugged in screen mode at a time. The remaining tasks must be debugged in line mode.

User Action: Continue debugging in line mode. If you exit from screen mode in the other task, you can use screen mode in the current task.

--ERROR DB 270-- Errors encountered while trying to regenerate the source for {text}.

Condition Identifier: DBE\$FS_ERROR_DURING_REGEN Code: DB 270

Description: The errors listed in the Output window were encountered while regenerating the source for the specified module. The errors are severe enough that no source can be made available for screen mode debugging for this module.

User Action: If the errors indicate that file is missing or damaged and if you can re-load or re-construct the file, press the ZMIN function to regenerate the source again.

--ERROR DB 271-- Debug abort - {text} at {text}{text}

Condition Identifier: DBE\$DEBUG_ABORT_WITH_TRACE Code: DB 271

Description: This is a Debug internal error.

User Action: Report this error to your system analyst.

--ERROR DB 272-- An S register value must be specified as integer.

Condition Identifier: DBE\$INVALID_SREG_VALUE Code: DB 272

Description: You can only change the value of a state register to an integer value.

User Action: Specify an integer value.

--WARNING DB 273-- No parameters were specified for the change_user_mask command.

Condition Identifier: DBE\$NO_PARMS_SPECIFIED Code: DB 273

Description: All parameters on this command are independent of each other and are optional.

User Action: None.

--ERROR DB 274-- The specified register value is out of range.

Condition Identifier: DBE\$ILLEGAL_REG_NUM Code: DB 274

Description: The value you specified for the X or A register is not in the correct range.

User Action: Specify 0 through 15 for X or A registers.

--ERROR DB 275-- The specified S register is undefined.

Condition Identifier: DBE\$UNDEFINED_S_REG Code: DB 275

Description: You have specified an undefined state register.

User Action: Specify a valid state register.

--ERROR DB 276-- The NUMBER parameter is required.

Condition Identifier: DBE\$NUM_PARM_REQ Code: DB 276

Description: No register number was specified for the \$REGISTER function. You must specify a register number for A, X or S registers.

User Action: Specify a register number.

Further Information: To see a list of defined state registers, see the Virtual State Hardware Reference manual.

--ERROR DB 277-- The integer {text} is too large. Unsigned integers must be less than 264.**

Condition Identifier: DBE\$UNSIGNED_INT_TO_LARGE Code: DB 277

Description: You cannot change the value of a C unsigned integer variable to a value greater than the maximum allowed.

User Action: Specify a value in the valid range.

--ERROR DB 278-- The GOTO command is not available until after the task has begun.

Condition Identifier: DBE\$GOTO_BEFORE_TASK Code: DB 278

Description: Debug was called before the task was defined. You cannot skip this initialization.

User Action: You can set a break on one of the first lines in your program. Then use GOTO when the break is reached.

--ERROR DB 279-- The GOTO command is not available after task termination.

Condition Identifier: DBE\$GOTO_AFTER_TASK Code: DB 279

Description: You cannot restart a terminated task.

User Action: To continue debugging, you must quit and reexecute your program in debug mode.

--ERROR DB 280-- You cannot use GOTO to transfer to this line.

Condition Identifier: DBE\$GOTO_TRANSFER Code: DB 280

Description: This statement is not executable, or its code has been moved by optimization, or it is not the first line of a multi-line statement.

User Action: Specify an executable statement.

--ERROR DB 281-- You cannot use GOTO to transfer out of the current procedure.

Condition Identifier: DBE\$GOTO_TRANSFER_OUT Code: DB 281

Description: You cannot use GOTO to transfer execution out of the current procedure.

User Action: Specify a line in the current procedure.

--ERROR DB 282-- The source regenerated using the source map for {text} contains source for module {text}.

Condition Identifier: DBE\$REGEN_GOT_WRONG_MODULE Code: DB 282

Description: The source map for the specified module contains information for a different module. Therefore, the source map cannot be used for this module.

User Action: To generate the correct source map, recompile the program. If the program is stored in an SCU deck, specify the DEBUG_AIDS=DEBUG_TABLES on the SCU EXPAND_DECK command. Then specify the INPUT_SOURCE_MAP on the compile command.

--ERROR DB 284-- The command_string parameter must be a string of commands or a function key short label.

Condition Identifier: DBE\$ILLEGAL_CS_ENTRY Code: DB 284

Description: An illegal value was entered for the SET_FUNCTION_KEY COMMAND_STRING parameter.

User Action: Enter a string of commands or a function key label name for the COMMAND_STRING parameter.

--ERROR DB 285-- In full screen mode, {text} is only legal in a command string associated with a break.

Condition Identifier: DBE\$ILLEGAL_FS_CMD_XCEPT_IN_CS Code: DB 285

Description: The RUN command cannot be used:

- in an INCLUDE_FILE
- in a user-defined function key command
- in screen mode, except in a command
- in a string associated with a break

User Action: Do not use the RUN command in these situations.

--ERROR DB 999-- {text}

Condition Identifier: DBE\$UNUSED_ERROR_CODE Code: DB 999

Description: Error message 640510 designates the last position available in the error message deck and is not referenced by Debug.

User Action: None.

--INFORMATIVE DB 1001-- {text} break {text}, {text} at {text}{text}

Condition Identifier: DBE\$USER_DEFINED_BREAK Code: DB 1001

Description: Part of the Debug informative text indicating where Debug received control.

User Action: None.

--INFORMATIVE DB 1002-- {text}

Condition Identifier: DBE\$PARMS_ONLY_MESSAGE Code: DB 1002

Description: A parameter which is not supported for this language was entered.

User Action: Verify the specified parameter conforms to this language.

--INFORMATIVE DB 1003-- Trap deferred from{text}{text}{text} {text}{text}

Condition Identifier: DBE\$DEFERRED_BREAK Code: DB 1003

Description: A trap has occurred at a more privileged location, and has been reported when control returns to the user's privileged ring.

User Action: None.

--INFORMATIVE DB 1004-- {text} program terminated by calling {text} at {text}{text}

Condition Identifier: DBE\$TERMINATION_MESSAGE Code: DB 1004

Description: The user's program has run to termination by calling indicated location.

User Action: None.

--INFORMATIVE DB 1005-- {text} QUIT terminated task

Condition Identifier: DBE\$QUIT_MESSAGE Code: DB 1005

Description: The QUIT command has terminated the user's task. The DEBUG session has completed.

User Action: None.

--INFORMATIVE DB 1006-- {text} {text} at {text}{text}

Condition Identifier: DBE\$UNSELECTED_EVENT Code: DB 1006

Description: Part of the Debug informative text indicating where Debug received control.

User Action: None.

--INFORMATIVE DB 1007-- {text} {text} address={text} at {text}{text}

Condition Identifier: DBE\$SAC Code: DB 1007

Description: Part of the display for Segment Access Conditions.

User Action: None.

--INFORMATIVE DB 1008-- {text} user_defined_condition {text} at {text}{text}

Condition Identifier: DBE\$USER_DEFINED_CONDITION Code: DB 1008

Description: The user-defined condition has occurred at the indicated location.

User Action: None.

--INFORMATIVE DB 1009-- {text} program terminated by returning

Condition Identifier: DBE\$PROGRAM_COMPLETED Code: DB 1009

Description: The program has run to completion.

User Action: None.

--INFORMATIVE DB 1010-- {text}{text}{text} {text}{text} {text}{text}{text}

Condition Identifier: DBE\$FORMAT_TRACEBACK_MESSAGE Code: DB 1010

Description: Part of the DISPLAY_CALLS display.

User Action: None.

--INFORMATIVE DB 1011-- Break {text}

Condition Identifier: DBE\$DISPLAY_BREAK_NAME Code: DB 1011

Description: Part of the DISPLAY_BREAK display.

User Action: None.

--INFORMATIVE DB 1012-- event(s) = {text} {text}

Condition Identifier: DBE\$DISPLAY_BREAK_EVENT Code: DB 1012

Description: Message is part of DISPLAY_BREAK text.

User Action: None.

--INFORMATIVE DB 1013-- no breaks set

Condition Identifier: DBE\$NO_BREAKS_SET Code: DB 1013

Description: Informative message that indicates that no breaks are set in the current DEBUG session.
Message is part of the DISPLAY_BREAK display.

User Action: None.

--INFORMATIVE DB 1014-- Break {text} not found

Condition Identifier: DBE\$UNKNOWN_BREAK_NAME Code: DB 1014

Description: The specified break could not be located.

User Action: Use the DISPLAY_BREAK, ALL command to view all currently active breaks.

--INFORMATIVE DB 1015-- range{text} {text}{text} {text}{text}

Condition Identifier: DBE\$BREAK_RANGE Code: DB 1015

Description: Range of the displayed break in the DISPLAY_BREAK output.

User Action: None.

--INFORMATIVE DB 1016-- location{text} {text}{text} {text}{text}

Condition Identifier: DBE\$BREAK_LOCATION Code: DB 1016

Description: Break location message for DISPLAY_BREAK output.

User Action: Not applicable.

--INFORMATIVE DB 1017-- command(s) = {text}

Condition Identifier: DBE\$COMMAND_STRING Code: DB 1017

Description: Part of the DISPLAY_BREAK display.

User Action: None.

--INFORMATIVE DB 1018-- eoi reached on Debug input file {text}, user task resumed

Condition Identifier: DBE\$RESUME_EOI_ON_INPUT Code: DB 1018

Description: The debug-input file contains no more commands.. The user task has been resumed.

User Action: None.

--INFORMATIVE DB 1019-- break {text} unknown and cannot be deleted.

Condition Identifier: DBE\$UNDELETABLE_BREAK Code: DB 1019

Description: Informative message stating that the break specified on the DELETE_BREAK command cannot be deleted because no break by that name exists.

User Action: Check that the correct name was specified on the DELETE_BREAK command. If needed, display the breaks with a DISPLAY_BREAKS command.

--INFORMATIVE DB 1020-- Default module is {text}.

Condition Identifier: DB\$DEFAULT_MODULE Code: DB 1020

Description: The text is part of the DISDE display.

User Action: None.

--INFORMATIVE DB 1021-- Default procedure is {text}.

Condition Identifier: DB\$DEFAULT_PROCEDURE Code: DB 1021

Description: The text is part of the DISDE display.

User Action: None.

--INFORMATIVE DB 1022-- Default debug_input file is {text}.

Condition Identifier: DB\$DEBUG_INPUT Code: DB 1022

Description: The text is part of the DISDE display.

User Action: None.

--INFORMATIVE DB 1023-- A change to {text} for debug_input is pending.

Condition Identifier: DB\$PENDING_DEBUG_INPUT Code: DB 1023

Description: The change to the debug_input file does not become effective until the RUN command is issued.

User Action: None.

--INFORMATIVE DB 1024-- Default debug_output file is {text}.

Condition Identifier: DB\$DEBUG_OUTPUT Code: DB 1024

Description: The text is part of the DISDE display.

User Action: None.

--INFORMATIVE DB 1025-- The number of breaks set by the user is{text}.

Condition Identifier: DB\$USER_BREAKS Code: DB 1025

Description: DISDE display line.

User Action: None.

--INFORMATIVE DB 1026-- The number of breaks in use by DEBUG is{text}.

Condition Identifier: DB\$INTERNAL_BREAKS_SET Code: DB 1026

Description: DISDE display line.

User Action: None.

--INFORMATIVE DB 1027-- The number of available breaks is{text}.

Condition Identifier: DB\$AVAILABLE_BREAKS Code: DB 1027

Description: Message is displayed in debugging environment. Message is one of the display lines indicating the number of breaks that the user can set in combination with DEBUG's internal breaks.

User Action: Not applicable.

--INFORMATIVE DB 1028-- Execution is currently stopped at {text} which, in symbolic terms is {text}{text}

Condition Identifier: DB\$USER_ADDRESS Code: DB 1028

Description: Part of Display_debugging_environment display.

User Action: None.

--INFORMATIVE DB 1029-- {text} step at {text}{text}

Condition Identifier: DB\$REPORT_STEP Code: DB 1029

Description: This information message lists STEP mode parameters.

User Action: None.

--INFORMATIVE DB 1030-- Step_mode is OFF.

Condition Identifier: DB\$STEP_MODE_OFF Code: DB 1030

Description: Informative text from the DISDE display.

User Action: None.

--INFORMATIVE DB 1031-- Step mode is ON{text}{text}, stepping by {text} in {text} {text}

Condition Identifier: DB\$STEP_MODE_ON Code: DB 1031

Description: Part of the DISDE display. The informative text indicates the current step-mode parameters in effect.

User Action: None.

--INFORMATIVE DB 1032-- DISPLAY OF ALL {text} IN {text}

Condition Identifier: DB\$DISPLAY_ALL_NAMES_HDR1 Code: DB 1032

Description: Message is header for DISPV \$ALL display.

User Action: None.

--INFORMATIVE DB 1033-- {text} is not active. Only static variables can be displayed.

Condition Identifier: DB\$DISPLAY_ALL_NAMES_HDR2 Code: DB 1033

Description: Display_all_names (DISPV \$ALL) informative message.

User Action: None.

--INFORMATIVE DB 1034-- No {text} found.

Condition Identifier: DB\$NO_VARS_FOUND Code: DB 1034

Description: No variables were located for the specified Display request.

User Action: Not applicable.

--INFORMATIVE DB 1035-- {text} The status at termination was: {text}

Condition Identifier: DB\$TERMINATION_STATUS Code: DB 1035

Description: This information message gives the status of the user's program at termination.

User Action: None.

--INFORMATIVE DB 1036-- Break name {text} assigned to this break

Condition Identifier: DB\$DEFAULT_BREAK_NAME Code: DB 1036

Description: The indicated break name is assigned by DEBUG to this break.

User Action: None.

--INFORMATIVE DB 1037-- {text}

Condition Identifier: DB\$STEP_MODE_QUALIFIERS Code: DB 1037

Description: Part of the SET_STEP_MODE display.

User Action: None.

--INFORMATIVE DB 1038-- Unable to access {text}. The current stack frame is not yet initialized.

Condition Identifier: DB\$IN_PROLOG_CODE Code: DB 1038

Description: Debug cannot access program information before the compiler's prolog code is initialized.

User Action: Not applicable.

--INFORMATIVE DB 1039-- DISPLAY OF ALL MODULE LEVEL {text} IN {text}

Condition Identifier: DBE\$DISPLAY_ALL_NAMES_HDR3 Code: DB 1039

Description: Message is header for DISPV \$ALL display.

User Action: None.

--INFORMATIVE DB 1040-- Execution is currently stopped at {text}.

Condition Identifier: DBE\$UNFORMATTABLE_USER_ADDR Code: DB 1040

Description: Part of DISPLAY_DEBUGGING_ENVIRONMENT display.

User Action: None.

--INFORMATIVE DB 1041-- The formatted status message is:

Condition Identifier: DBE\$FORMATTED_STATUS_IS Code: DB 1041

Description: The formatted status is displayed.

User Action: Not applicable.

--INFORMATIVE DB 1042-- DISPLAY OF ALL {text} IN SECTION {text} OF {text}

Condition Identifier: DBE\$DISPLAY_ALL_NAMES_HDR4 Code: DB 1042

Description: Message header for DISPLAY_PROGRAM_VALUE \$ALL SECTION=name.

User Action: None.

--INFORMATIVE DB 1043-- {text} user_defined_condition text follows:

Condition Identifier: DBE\$CYBIL_RUNTIME_CONDITION Code: DB 1043

Description: Informative text displayed when a CYBIL runtime error is encountered in a multi-tasking session.

User Action: None.

--INFORMATIVE DB 1044-- Status: Terminated {text} {text}{text}

Condition Identifier: DBE\$TASK_TERMINATION_STATUS Code: DB 1044

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1045-- Status: {text}

Condition Identifier: DBE\$TASK_STATUS Code: DB 1045

Description: Message for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1046-- TASK {text}: system id: {text}

Condition Identifier: DBE\$TASK_NUMBER Code: DB 1046

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1047-- Terminated {text}

Condition Identifier: DBE\$TERMINATION_REASON Code: DB 1047

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1048-- Currently active task(s) -{text}

Condition Identifier: DBE\$NUM_ACTIVE Code: DB 1048

Description: Message text for the output of DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1049-- Transfer address is {text}{text}{text} {text}{text}

Condition Identifier: DBE\$TRNSFR_ADDR Code: DB 1049

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1050-- Total task(s) activated -{text}

Condition Identifier: DBE\$NUM_ACTIVATED Code: DB 1050

Description: Message text for the output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1051-- Stopped {text} at {text}{text}

Condition Identifier: DBE\$IS_STOPPED Code: DB 1051

Description: Message text for DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1052-- Last stopped {text} at {text}{text}

Condition Identifier: DBE\$WAS_STOPPED Code: DB 1052

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1053-- Reason for stopping is unknown

Condition Identifier: DBE\$UNKNOWN_STOPPAGE Code: DB 1053

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1054-- Queued commands: {text}

Condition Identifier: DBE\$QUEUED_CMDS Code: DB 1054

Description: Message text for Output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1055-- Incorrect value(s) specified for the task number parameter: {text}

Condition Identifier: DBE\$BAD_TASKS Code: DB 1055

Description: Some, but not all, of the values specified for the TASK_NUMBER parameter represent unknown tasks. The message contains all that are incorrect. This allows the command to go ahead and execute if at least one task number is entered correctly.

User Action: None.

--INFORMATIVE DB 1056-- Style: {text} {text}

Condition Identifier: DBE\$STYLE Code: DB 1056

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1057-- More tasks have been activated than can be displayed

Condition Identifier: DBE\$FILE_OVERFLOW Code: DB 1057

Description: The number of tasks that can be displayed has exceeded the maximum. Therefore, not all task information has been displayed.

User Action: None.

--INFORMATIVE DB 1058-- The commands which follow are queued but cannot be executed: {text}

Condition Identifier: DBE\$CANT_EXECUTE_CMDS Code: DB 1058

Description: Reserved for future implementation.

User Action: None.

--INFORMATIVE DB 1059-- {text} {text} watch at {text}{text}

Condition Identifier: DBE\$REPORT_WATCH Code: DB 1059

Description: Message text for the Watch log. The log is specified in the modify window. The log is written to the Output window.

User Action: None.

--INFORMATIVE DB 1060-- {text} = {text}

Condition Identifier: DBE\$LOG_WATCH Code: DB 1060

Description: Message text for the Watch log that is displayed in the Output window.

User Action: None.

--INFORMATIVE DB 1061-- {text} User directed change of Watched variable at {text}{text}

Condition Identifier: DBE\$USER_DIRECTED_CHANGE Code: DB 1061

Description: The value of the variable being watched has been changed by a CHAPV or a CHAM command.

User Action: None.

--INFORMATIVE DB 1062-- Not enough breaks remain to restore all the Watched variables.

Condition Identifier: DBE\$CANT_RECOVER_WATCH_BREAKS Code: DB 1062

Description: When DEAS is entered, Watch definitions are saved. They are restored when ACTS is entered. However, not enough hardware break definitions exist to restore all of the Watches.

User Action: You can press WATCH to discover which were restored. You can use CANCEL to cancel some so that others can be reentered. You can use the DELETE_BREAK line command to delete line mode breaks that may have caused the maximum number of breaks to be reached.

--INFORMATIVE DB 1063-- Waiting for task {text} to complete.

Condition Identifier: DBE\$WAITING_FOR_OTHER_TASK Code: DB 1063

Description: Message text for output of the DISPLAY_DEBUG_TASK_STATUS command.

User Action: None.

--INFORMATIVE DB 1064-- The Watch on {text} conflicted with the Watch on {text}. The Watch on {text} was cancelled.

Condition Identifier: DBE\$CONFLICTING_WATCHES Code: DB 1064

Description: You cannot watch the same variable more than once; that is, the same address, not necessarily the same name when:

- Read and/or Write has been specified for the same address
- Monitor has been specified and the same name appears in the same procedure of the same module.

User Action: None. Debug has already deleted the second variable.

--INFORMATIVE DB 1065-- You cannot Watch a variable of this type. The Watch on {text} was cancelled.

Condition Identifier: DB\$WRONG_TYPE_FOR_WATCH Code: DB 1065

Description: You cannot watch the following:

- BASIC strings or arrays, unless you specify monitor
- Constant
- FORTRAN assumed size or assumed shape arrays
- FORTRAN array sections
- Variables with length of bits or length of zero
- Substring or subscripts that use non-literal qualifiers

User Action: None. Debug has cancelled the watch.

--INFORMATIVE DB 1066-- You cannot Watch an assumed size array. The Watch on {text} was cancelled.

Condition Identifier: DB\$CANT_WATCH_ASSUMED_SIZE Code: DB 1066

Description: You cannot watch this kind of array because its exact size is not known.

User Action: You can use SEEVAL or the DISPLAY_PROGRAM_VALUE command to display the actual array in the calling program. (The assumed-size array is a parameter in the called program.)

--INFORMATIVE DB 1067-- You cannot Watch an array section. The Watch on {text} was cancelled.

Condition Identifier: DB\$CANT_WATCH_ARRAY_SEC Code: DB 1067

Description: Watches are not allowed on array sections because they may not represent one contiguous region of memory.

User Action: You can watch an entire array or watch a series of individual elements. Enter the array elements one at a time.

--INFORMATIVE DB 1068-- A variable that is bit-aligned or whose size is measured in bits cannot be Watched. The Watch on {text} was cancelled.

Condition Identifier: DB\$CANT_WATCH_BITS Code: DB 1068

Description: You cannot select the following types of variables for the Watch display:

- A variable that is bit-aligned
- A variable whose size is measured in bits.

User Action: You can use SeeVal or the DISPLAY_PROGRAM_VALUE command to see the current values of these kinds of variables.

--INFORMATIVE DB 1069-- You cannot Watch a constant. The Watch on {text} was cancelled.

Condition Identifier: DB\$CANT_WATCH_CONSTANT Code: DB 1069

Description: Only variables can be watched in the Watch Display.

User Action: None.

--INFORMATIVE DB 1070-- Non-literal qualifiers cannot be used to define a Watched variable. The Watch on {text} was cancelled.

Condition Identifier: DB\$CANT_WATCH_WITH_NONLITERAL Code: DB 1070

Description: You cannot use such qualifiers to define substrings or array elements to be watched because it does not represent the same variable each time Debug is called.

User Action: Specify substrings and array elements using constant subscripts or substring references. For example, in FORTRAN, use X(1), not X(I).

--INFORMATIVE DB 1071-- A variable whose length is zero cannot be Watched. The Watch on {text} was cancelled.

Condition Identifier: DBE\$CANT_WATCH_WHEN_LEN_ZERO Code: DB 1071

Description: Since the variable is of length zero, there is nothing to watch.

User Action: None.

--INFORMATIVE DB 1072-- Not enough breaks are left to setup the Watch on {text}. This Watch was cancelled.

Condition Identifier: DBE\$NEED_MORE_BRKS_FOR_WATCH Code: DB 1072

Description: The specified Watch used Kind=Read and/or Write. This requires at least one break, but the maximum number of breaks has been reached.

User Action: Use CANCEL to delete some Watches whose Kind is Read and/or Write. Use DELETE_BREAK to delete line mode breaks with events of Read_Next_Instruction, Branch, Call, Read, or Write. Specify this variable to be watched in Kind=Monitor.

--INFORMATIVE DB 1073-- The Watch on {text} could not be setup because it overlapped with break {text}. This Watch was cancelled.

Condition Identifier: DBE\$WATCH_OVERLAP_WITH_BRK Code: DB 1073

Description: A Read/Write watch may not overlap with a Read/Write break set by the SET_BREAK command for the same address.

User Action: None. Debug has cancelled the watch.

--INFORMATIVE DB 1074-- Break {text} was not set because it overlapped with the Watch of {text}.

Condition Identifier: DBE\$BRK_OVERLAP_WITH_WATCH Code: DB 1074

Description: The Read or Write Watch of a variable is not allowed to overlap with a break set on the same variable. No break has been set.

User Action: To set the break, you must first cancel the Watch of the variable specified in the message.

--INFORMATIVE DB 1075-- The Watch on {text} was cancelled.

Condition Identifier: DBE\$WATCH_AUTO_CANCELLED Code: DB 1075

Description: The watch on the specified variable has been cancelled.

User Action: The previous message displayed explains why the watch was cancelled.

--INFORMATIVE DB 1256-- {text}

Condition Identifier: DBE\$UNUSED_INFORMATIVE_CODE Code: DB 1256

Description: Error message 641255 designates the last position available in the information message deck and is not referenced by Debug.

User Action: None.

--INFORMATIVE DB 2001-- -- Welcome to Full Screen Debugging --

Condition Identifier: DBE\$FS_WELCOME Code: DB 2001

Description: Full-screen header.

User Action: None.

--INFORMATIVE DB 2002-- Press {text} for assistance

Condition Identifier: DBE\$FS_HELP_AVAILABLE Code: DB 2002

Description: Full-screen instructive text.

User Action: Follow instructions in message.

--INFORMATIVE DB 2003-- Debugging {text}

Condition Identifier: DBE\$FS_DEBUGGING_HEADER Code: DB 2003

Description: Message is header for Zoom-in display for Full-screen Debug.

User Action: None.

--INFORMATIVE DB 2004-- Displaying Routines

Condition Identifier: DBE\$FS_DISPLAYING_HEADER Code: DB 2004

Description: Full-screen message is header for Zoom-out display.

User Action: None.

--INFORMATIVE DB 2005-- Enter number of lines to step

Condition Identifier: DBE\$FS_MULTIPLE_STEP_PROMPT Code: DB 2005

Description: Full-screen instructive text.

User Action: Enter the requested information.

--INFORMATIVE DB 2006-- Stepping {text}

Condition Identifier: DBE\$FS_STEP_COUNT Code: DB 2006

Description: Full-screen informative text. Step mode is currently stepping by the specified count.

User Action: None.

--INFORMATIVE DB 2007-- Enter name(s) to see

Condition Identifier: DBE\$FS_SEE_VALUE_PROMPT Code: DB 2007

Description: Full-screen instructive text requests on SEE value.

User Action: Enter requested name.

--INFORMATIVE DB 2008-- Enter a program variable name

Condition Identifier: DBE\$FS_SEE_VALUE_ERR1 Code: DB 2008

Description: Full-screen instructive text. A prompt to enter a program variable name for the SEE_VALUE command.

User Action: Enter a program variable name.

--INFORMATIVE DB 2009-- Enter name=value

Condition Identifier: DBE\$FS_CHANGE_VALUE_PROMPT Code: DB 2009

Description: Full-screen instructive text.

User Action: Enter the requested information.

--INFORMATIVE DB 2010-- Enter name=value form (e.g., x=4.3)

Condition Identifier: DBE\$FS_CHANGE_VALUE_ERR1 Code: DB 2010

Description: Full-screen instructive text.

User Action: Enter the requested information.

--INFORMATIVE DB 2011-- Move cursor to a program line before setting break

Condition Identifier: DBE\$FS_SET_BRK_ERR1 Code: DB 2011

Description: Full-screen set break error.

User Action: Not applicable.

--INFORMATIVE DB 2012-- Break already set on this line

Condition Identifier: DBE\$FS_SET_BRK_ERR2 Code: DB 2012

Description: Only one break is allowed per line.

User Action: Set a break on another line.

--INFORMATIVE DB 2013-- Limit is 64 breaks

Condition Identifier: DBE\$FS_SET_BRK_ERR3 Code: DB 2013

Description: Full-screen message. Only 32 breaks can be set.

User Action: Use CLR BRK to remove unnecessary breaks.

--INFORMATIVE DB 2014-- Select a break to clear

Condition Identifier: DBE\$FS_CLR_BRK_ERR1 Code: DB 2014

Description: Full-screen instructive text asks for a name of a break to clear.

User Action: Give name of break to be cleared.

--INFORMATIVE DB 2015-- No full screen break on the selected line

Condition Identifier: DBE\$FS_CLR_BRK_ERR2 Code: DB 2015

Description: Full-screen message. The CLR BRK command cannot locate a break at this line.

User Action: Select a line where a break is currently set before using the CLR BRK command.

--INFORMATIVE DB 2016-- Enter text to locate

Condition Identifier: DBE\$FS_LOCT_TEXT_PROMPT Code: DB 2016

Description: Full-screen instructive text.

User Action: Enter the requested information.

--INFORMATIVE DB 2017-- "{text}" not found

Condition Identifier: DBE\$FS_LOCT_TEXT_ERR1 Code: DB 2017

Description: Full-screen message. The specified text could not be found.

User Action: Not applicable.

--INFORMATIVE DB 2018-- Put cursor on routine to view and press ZmIn

Condition Identifier: DBE\$FS_ZOOMOUT_PROMPT Code: DB 2018

Description: Full-screen instructive text.

User Action: Follow instructions in message.

--INFORMATIVE DB 2019-- Program terminated

Condition Identifier: DBE\$FS_NORMAL_TERMINATION Code: DB 2019

Description: Full-screen informative message. Program has terminated execution.

User Action: None.

--INFORMATIVE DB 2020-- Stepping stopped by other event

Condition Identifier: DBE\$FS_STEP_STOPPED_BY_BREAK Code: DB 2020

Description: Full-screen informative text. Step mode has stopped because another break event has occurred.

User Action: None.

--INFORMATIVE DB 2021-- OUTPUT

Condition Identifier: DBE\$FS_OUTPUT_WINDOW_HEADER Code: DB 2021

Description: Full-screen header message.

User Action: None.

--INFORMATIVE DB 2022-- {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text} {text}

Condition Identifier: DBE\$FS_LOC_ANNOUNCEMENT Code: DB 2022

Description: Full-screen message is used to format in symbolic terms where execution has stopped when no other message applies.

User Action: None.

--INFORMATIVE DB 2023-- Call Trace

Condition Identifier: DBE\$FS_CALL_TRACE_HEADER Code: DB 2023

Description: Full-screen informative message. Trace has been initiated.

User Action: None.

--INFORMATIVE DB 2024-- called from

Condition Identifier: DBE\$FS_TB_MSG1 Code: DB 2024

Description: Full-screen informative text displayed when Trace is initiated.

User Action: None.

--INFORMATIVE DB 2025-- Place the cursor on the call to see and press ZmIn

Condition Identifier: DBE\$FS_CALL_TRACE_PROMPT Code: DB 2025

Description: Full-screen instructive text.

User Action: Follow instructions in message.

--INFORMATIVE DB 2026-- {text} calls not displayed

Condition Identifier: DBE\$FS_OVERFLOW_ON_CALL_TRACE Code: DB 2026

Description: Full-screen informative text. Message appears when number of trace calls overflows the source window.

User Action: None.

--INFORMATIVE DB 2027-- There are no active calls

Condition Identifier: DBE\$FS_NO_ACTIVE_CALLS Code: DB 2027

Description: Full-screen informative text for TRACE.

User Action: None.

--INFORMATIVE DB 2028-- H E L P

Condition Identifier: DBE\$FS_HELP_HEADER Code: DB 2028

Description: Full-screen help header.

User Action: None.

--INFORMATIVE DB 2029-- For HELP with a DEBUG function, press the corresponding function key.

Condition Identifier: DBE\$FS_HELP_PROMPT Code: DB 2029

Description: Full-screen message is displayed in the HELP window.

User Action: None.

--INFORMATIVE DB 2030-- Press NEXT to return to debugging.

Condition Identifier: DBE\$FS_HELP_EXIT_PROMPT Code: DB 2030

Description: Full-screen help instructive text.

User Action: Follow instructions in message.

--INFORMATIVE DB 2031-- Press NEXT for more information.

Condition Identifier: DBE\$FS_HELP_OVERFLOW_PROMPT Code: DB 2031

Description: Full-screen help instructive text.

User Action: Follow instructions in message.

--INFORMATIVE DB 2032-- This function is currently disabled.

Condition Identifier: DBE\$FS_HELP_DISABLED_FUNCTION Code: DB 2032

Description: You requested HELP for a function that is not currently used.

User Action: None.

--INFORMATIVE DB 2033-- This function key ({text}) is not used by DEBUG.

Condition Identifier: DBE\$FS_HELP_UNDEFINED_FUNCTION Code: DB 2033

Description: Message is HELP description displayed when function is not assigned a function key.

User Action: None.

--INFORMATIVE DB 2034-- Step1 causes the program to execute until the next line is encountered.

Condition Identifier: DB\$FS_HELP_STEP1_FUNCTION Code: DB 2034

Description: Message is HELP description of STEP1 full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2035-- StepN causes the next 'n' lines of the program to be executed. DEBUG prompts for the value of n.

Condition Identifier: DB\$FS_HELP_STEPN_FUNCTION Code: DB 2035

Description: Message is HELP description of STEPN full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2036-- MSpeed causes the program to execute until the next routine or break is encountered.

Condition Identifier: DB\$FS_HELP_MSPEED_FTN_FUNCTION Code: DB 2036

Description: Full-screen help text - FORTRAN.

User Action: Not applicable.

--INFORMATIVE DB 2037-- MSpeed causes the program to execute until the next section or break is encountered.

Condition Identifier: DB\$FS_HELP_MSPEED_CBL_FUNCTION Code: DB 2037

Description: Full-screen help text - COBOL.

User Action: Not applicable.

--INFORMATIVE DB 2038-- LSpeed causes the program to execute until the next paragraph or break is encountered.

Condition Identifier: DB\$FS_HELP_LSPEED_FUNCTION Code: DB 2038

Description: Message is HELP description of LSPEED full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2039-- HSpeed causes the program to execute until the next break is encountered.

Condition Identifier: DB\$FS_HELP_HSPEED_FUNCTION Code: DB 2039

Description: Message is HELP description of HSPEED full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2040-- Use the Locate function to find a string of characters in the current source module or zoom out table. Locate will not cross source module boundaries.

Condition Identifier: DB\$FS_HELP_LOCATE_FUNCTION Code: DB 2040

Description: Message is HELP description of LOCATE full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2041-- SeeVal will display the current value of a variable in the output window.

Condition Identifier: DB\$FS_HELP_SEEVAL_FUNCTION Code: DB 2041

Description: Message is HELP description of SEEVAL full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2042-- ChaVal will change the current value of a variable. DEBUG prompts for the new value.

Condition Identifier: DBE\$FS_HELP_CHGVAL_FUNCTION Code: DB 2042

Description: Full-screen message displaying help information for CHAVAL function.

User Action: None.

--INFORMATIVE DB 2043-- SetBrk sets a break on the source line where the cursor is currently positioned.

Condition Identifier: DBE\$FS_HELP_SETBRK_FUNCTION Code: DB 2043

Description: Message is HELP description of SETBRK full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2044-- DelBrk deletes any break set at the current cursor position.

Condition Identifier: DBE\$FS_HELP_DELBRK_FUNCTION Code: DB 2044

Description: Full-screen message displaying help information for DELBRK function.

User Action: None.

--INFORMATIVE DB 2045-- Quit terminates the current DEBUG session.

Condition Identifier: DBE\$FS_HELP_QUIT_FUNCTION Code: DB 2045

Description: Message is HELP description of QUIT full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2046-- Trace provides a call traceback list.

Condition Identifier: DBE\$FS_HELP_TRACE_FUNCTION Code: DB 2046

Description: Message is HELP description of TRACE full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2047-- ZmOut displays the routines in the source program. ZmIn returns to the line level.

Condition Identifier: DBE\$FS_HELP_ZOOM_FTN_FUNCTION Code: DB 2047

Description: Full-screen help text - FORTRAN.

User Action: Not applicable.

--INFORMATIVE DB 2048-- ZmOut displays the paragraphs and sections in the source program. ZmIn returns to the line level.

Condition Identifier: DBE\$FS_HELP_ZOOM_CBL_FUNCTION Code: DB 2048

Description: Full-screen help text - COBOL.

User Action: Not applicable.

--INFORMATIVE DB 2049-- FWD pages the window containing the cursor forward. Both the source and output windows can be paged. FWD from the last screen of a source module goes to the first screen of the next initialized source module.

Condition Identifier: DBE\$FS_HELP_FWD_FUNCTION Code: DB 2049

Description: Full-screen message displaying help information for FWD function.

User Action: None.

--INFORMATIVE DB 2050-- BKW pages the window containing the cursor backward. Both the source and output windows can be paged. BKW from the first screen of a source module goes to the last screen of the previous initialized source module.

Condition Identifier: DBE\$FS_HELP_BKW_FUNCTION Code: DB 2050

Description: Full-screen message displaying help information for BKW function.

User Action: None.

--INFORMATIVE DB 2051-- BACK usually returns to a previous state. E.g., BACK is equivalent to a ZmIn after a ZmOut.

Condition Identifier: DBE\$FS_HELP_BACK_FUNCTION Code: DB 2051

Description: Full-screen message displaying help information for BACK function.

User Action: None.

--INFORMATIVE DB 2052-- UP moves the line containing the cursor to the top of the window.

Condition Identifier: DBE\$FS_HELP_UP_FUNCTION Code: DB 2052

Description: Message is HELP description of UP full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2053-- DOWN moves the line containing the cursor to the bottom of the window.

Condition Identifier: DBE\$FS_HELP_DOWN_FUNCTION Code: DB 2053

Description: Full-screen message displaying help information for DOWN function.

User Action: None.

--INFORMATIVE DB 2054-- High speed

Condition Identifier: DBE\$FS_HIGH_SPEED Code: DB 2054

Description: Full-screen high speed header.

User Action: None.

--INFORMATIVE DB 2055-- Enter input requested below

Condition Identifier: DBE\$FS_INPUT_REQUESTED Code: DB 2055

Description: Full-screen instructive text.

User Action: Enter the requested input.

--INFORMATIVE DB 2056-- Low speed

Condition Identifier: DBE\$FS_LOW_SPEED Code: DB 2056

Description: Full-screen header message.

User Action: None.

--INFORMATIVE DB 2057-- Medium speed

Condition Identifier: DBE\$FS_MEDIUM_SPEED Code: DB 2057

Description: Full-screen header message.

User Action: None.

--INFORMATIVE DB 2058-- Goto resumes execution of the program at the line containing the cursor.

Condition Identifier: DBE\$FS_HELP_GOTO_FUNCTION Code: DB 2058

Description: Full-screen message displaying help information for GOTO function.

User Action: None.

--INFORMATIVE DB 2059-- Move cursor to a line in the source program before using {text}

Condition Identifier: DB\$FS_GOTO_ERR1 Code: DB 2059

Description: Full-screen message. The cursor must be located on a line in the source program of the source window before the GOTO function is pressed.

User Action: Move the cursor to a line in the source window, then press GOTO.

--INFORMATIVE DB 2060-- You may not use {text} to transfer out of the current routine

Condition Identifier: DB\$FS_GOTO_ERR2 Code: DB 2060

Description: Full-screen message. The specified GOTO location could not be honored.

User Action: Not applicable.

--INFORMATIVE DB 2061-- This is not an allowed input

Condition Identifier: DB\$FS_UNALLOWED_INPUT Code: DB 2061

Description: The specified line mode Debug command cannot be entered on the HOME line in screen mode.

User Action: If the desired goal cannot be accomplished with function keys, use DEAS to move to line. Then reenter the line mode command.

--INFORMATIVE DB 2062-- This command input is too long

Condition Identifier: DB\$FS_COMMAND_TOO_LONG Code: DB 2062

Description: The text entered on the HOME line, including continuation lines, exceeds the maximum length allowed by full screen Debug.

User Action: Break the input text into several separate Debug inputs.

--INFORMATIVE DB 2063-- Keys shows you a list of the currently available debugging functions.

Condition Identifier: DB\$FS_HELP_KEYS_FUNCTION Code: DB 2063

Description: Message is HELP description of KEYS full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2064-- Screen functions and Debug commands may be entered on the home line. Available screen functions are:

Condition Identifier: DB\$FS_KEYS_HEADER Code: DB 2064

Description: Full-screen message is header for KEYS window.

User Action: None.

--INFORMATIVE DB 2067-- ERROR - Look for text in output window

Condition Identifier: DB\$FS_ERROR_NOTIFICATION Code: DB 2067

Description: The message is too long to display in the response line. The complete message is displayed in the output window.

User Action: None.

--INFORMATIVE DB 2068-- HOME moves the cursor to the home line for input of commands. DEBUG function keys are disabled while commands are being entered in this way.

Condition Identifier: DB\$FS_HELP_HOME_FUNCTION Code: DB 2068

Description: Message is HELP description of HOME full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2069-- Refrsh erases the screen and repaints it with current information.

Condition Identifier: DB\$FS_HELP_CLEAR_FUNCTION Code: DB 2069

Description: Full-screen message displaying help information for REFRSH function.

User Action: None.

--INFORMATIVE DB 2070-- A break cannot be set at this line

Condition Identifier: DB\$FS_SET_BRK_ERR4 Code: DB 2070

Description: The user cannot set a break at the selected line. (Not all lines in a source program can be selected for a break condition.)

User Action: Select another line for a break.

--INFORMATIVE DB 2071-- DEBUG: {text} at {text}

Condition Identifier: DB\$FS_UNSELECTED_EVENT Code: DB 2071

Description: Full-screen message header.

User Action: None.

--INFORMATIVE DB 2072-- Overflow of response line message - See Output Window

Condition Identifier: DB\$FS_RESPONSE_LINE_OVERFLOW Code: DB 2072

Description: Full-screen informative text. Message appears when text overflows response line. Instead, text is displayed in output window.

User Action: None.

--INFORMATIVE DB 2073-- Enter more text

Condition Identifier: DB\$FS_TEXT_OVERFLOW_PROMPT Code: DB 2073

Description: Full-screen instructive text. Appears when text overflows the input field.

User Action: Enter the text that overflowed the input field.

--INFORMATIVE DB 2074-- Use Left in the source window to make the leftmost portion of the window visible.

Condition Identifier: DB\$FS_HELP_LEFT_FUNCTION Code: DB 2074

Description: Message is HELP description of LEFT full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2075-- Use Right in the source window to move the visible portion of the window to the right.

Condition Identifier: DB\$FS_HELP_RIGHT_FUNCTION Code: DB 2075

Description: Message is HELP description of RIGHT full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2076-- N must be a positive integer

Condition Identifier: DB\$FS_STEPN_ERR1 Code: DB 2076

Description: Full-screen informative text. The integer specified for stepping must be a positive integer.

User Action: None.

--INFORMATIVE DB 2077-- INPUT

Condition Identifier: DB\$FS_INPUT_WINDOW_HDR Code: DB 2077

Description: Full-screen message is header for INPUT window.

User Action: None.

--INFORMATIVE DB 2078-- Source for {text} was not found in file {text}

Condition Identifier: DBE\$FS_MODULE_NOT_IN_SOURCE Code: DB 2078

Description: The source specified is not currently displayed in the Zoom-in Display.

User Action: None.

--INFORMATIVE DB 2079-- Enter compiler input file for {text}

Condition Identifier: DBE\$FS_NEW_SOURCE_PROMPT Code: DB 2079

Description: Full-screen instructive text.

User Action: Enter the file name that contains the source code to be debugged.

--INFORMATIVE DB 2080-- The specified routine was compiled without DEBUG tables.

Condition Identifier: DBE\$FS_NO_DEBUG_TABLES Code: DB 2080

Description: You cannot select the Zoom-in display for a routine that has not been compiled with Debug tables.

User Action: Compile the routine with Debug tables.

--INFORMATIVE DB 2081-- First displays the first screen of the current source module, zoom out table, or output window. First will not cross source module boundaries.

Condition Identifier: DBE\$FS_HELP_FIRST_FUNCTION Code: DB 2081

Description: Full-screen message displaying help information for FIRST function.

User Action: None.

--INFORMATIVE DB 2082-- Last displays the last screen of the current source module, zoom out table, or output window. Last will not cross source module boundaries.

Condition Identifier: DBE\$FS_HELP_LAST_FUNCTION Code: DB 2082

Description: Message is HELP description of LAST full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2083-- Deas deactivates the full screen debugger. Line mode debugging is then available.

Condition Identifier: DBE\$FS_HELP_DEAS_FUNCTION Code: DB 2083

Description: Full-screen message displaying help information for DEAS function.

User Action: None.

--INFORMATIVE DB 2084-- Opts brings up the DEBUG options window.

Condition Identifier: DBE\$FS_HELP_OPTS_FUNCTION Code: DB 2084

Description: Message is HELP description of OPTS full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2085-- Split redefines the relative sizes of the source and output windows. The cursor position determines the new boundary between the two windows.

Condition Identifier: DBE\$FS_HELP_SPLIT_FUNCTION Code: DB 2085

Description: Message is HELP description of SPLIT full-screen function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2086-- Press matching menu item to toggle value.

Condition Identifier: DBE\$FS_OPTS_INSTRUCTION Code: DB 2086

Description: Full-screen instructive text is displayed in the OPTIONS window.

User Action: Press a function to toggle an option or press RETURN to remove the OPTIONS window.

--INFORMATIVE DB 2087-- DEBUG OPTIONS WINDOW

Condition Identifier: DB\$FS_OPTS_HEADER_TEXT Code: DB 2087

Description: Full-screen message is header for OPTIONS window.

User Action: None.

--INFORMATIVE DB 2088-- FEATURE(menu item)

Condition Identifier: DB\$FS_OPTS_SUBHEADER1 Code: DB 2088

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2089-- VALUE

Condition Identifier: DB\$FS_OPTS_SUBHEADER2 Code: DB 2089

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2090-- AVAILABLE OPTIONS

Condition Identifier: DB\$FS_OPTS_SUBHEADER3 Code: DB 2090

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2091-- MENU ROWS (F1)

Condition Identifier: DB\$FS_OPTS_MENU_ROWS Code: DB 2091

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2092-- MODE (F2)

Condition Identifier: DB\$FS_OPTS_MODE Code: DB 2092

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2093-- SOURCE SIZE (F3)

Condition Identifier: DB\$FS_OPTS_SOURCE_SIZE Code: DB 2093

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2094-- OUTPUT SIZE (F4)

Condition Identifier: DB\$FS_OPTS_OUTPUT_SIZE Code: DB 2094

Description: Full-screen informative message in OPTIONS window.

User Action: None.

--INFORMATIVE DB 2095-- Cannot switch to user mode when in a system module.

Condition Identifier: DB\$FS_OPTS_ILLEGAL_MODE_SWITCH Code: DB 2095

Description: You cannot select USER mode in the OPTIONS window when a system module is selected or being debugged.

User Action: Before switching to USER mode, select a USER module.

--INFORMATIVE DB 2096-- Execution has stopped in system module {text}.

Condition Identifier: DB\$FS_STOPPED_IN_SYSTEM_MOD Code: DB 2096

Description: Full-screen message informing you that execution has stopped in a system module for which no source is available.

User Action: None.

--INFORMATIVE DB 2098-- Wide displays 80 characters on the screen. Narrow displays 132 characters.

Condition Identifier: DBE\$FS_HELP_SCREEN_WIDTH_FUNC Code: DB 2098

Description: Full-screen informative message displayed in HELP window.

User Action: None.

--INFORMATIVE DB 2099-- The specified break overlaps with an existing line mode break.

Condition Identifier: DBE\$FS_SET_BRK_ERR5 Code: DB 2099

Description: There already exists a break at the line specified. The existing break was set before entering screen mode or was set with the SET_BREAK command on the home line. In either case, the break is not known to the full screen processor and is, therefore, not highlighted.

User Action: If you do not want the line mode break, you must delete it before the new break can be set.

--INFORMATIVE DB 2100-- The source may not be accurately represented for {text}.

Condition Identifier: DBE\$FS_WARNING_DURING_REGEN Code: DB 2100

Description: The warning errors listed in the Output window were encountered while regenerating the source for the specified module. The errors are minor enough so that you can choose to proceed even though the source may not match the binary exactly.

User Action: If you think the differences between the source and the binary are too great, this error can be eliminated by re-compiling the source for this module.

--INFORMATIVE DB 2101-- Press the HELP function for more information.

Condition Identifier: DBE\$FS_HELP_MORE_INFO_PROMPT Code: DB 2101

Description: Text displayed in Help window.

User Action: None.

--INFORMATIVE DB 2102-- Pressing any other function gives help for that function.

Condition Identifier: DBE\$FS_HELP_OTHER_FUNC_INFO Code: DB 2102

Description: Text displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2103-- Debug function key labels, most line mode Debug commands and commands to other active command processors may be entered on this line, the HOME line.

Condition Identifier: DBE\$FS_HELP_HOME_LINE Code: DB 2103

Description: Full-screen message displaying help information for the HOME line.

User Action: None.

--INFORMATIVE DB 2104-- Short responses and advisory messages from Debug are displayed on this line, the RESPONSE line.

Condition Identifier: DBE\$FS_HELP_RESPONSE_LINE Code: DB 2104

Description: Message is HELP description of the Response (message) line displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2105-- The zoomed-in display shows the source code of the module in the program being debugged.

Condition Identifier: DBE\$FS_HELP_SRC_WINDOW_ZMD_IN Code: DB 2105

Description: Message is HELP description of the Zoom-in display shown in the Help window.

User Action: None.

--INFORMATIVE DB 2106-- The zoomed-out display lists the modules in the program being debugged.

Condition Identifier: DB\$FS_HELP_SRC_WINDOW_ZMD_OUT Code: DB 2106

Description: Message is HELP description of the Zoom-out display shown in the Help window.

User Action: None.

--INFORMATIVE DB 2107-- The trace display shows the routines in the active call chain.

Condition Identifier: DB\$FS_HELP_TRACE Code: DB 2107

Description: Message is HELP description of the Trace display shown in the Help window.

User Action: None.

--INFORMATIVE DB 2108-- The output window is the area in which interactive line mode output generated by your code or output delivered by Debug is displayed.

Condition Identifier: DB\$FS_HELP_OUTPUT_WINDOW Code: DB 2108

Description: Message is HELP description of the Output window displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2109-- The watch window displays the current values of variables you would like tracked.

Condition Identifier: DB\$FS_HELP_WATCH_WINDOW Code: DB 2109

Description: Message is HELP description of the Watch display shown in the Help window.

User Action: None.

--INFORMATIVE DB 2111-- WATCH VALUE

Condition Identifier: DB\$FS_WATCH_VALUE_HDR Code: DB 2111

Description: Message header in Watch display.

User Action: None.

--INFORMATIVE DB 2112-- This variable cannot be Watched

Condition Identifier: DB\$FS_WRONG_TYPE_FOR_WATCH Code: DB 2112

Description: You cannot watch the following:

- BASIC strings or arrays, unless you specify Monitor
- A constant
- FORTRAN assumed-size or assumed-shape arrays
- FORTRAN array sections
- Variables with length of bits or length of zero
- Substring or subscripts that use non-literal qualifiers

User Action: Check to see that the variable is spelled and formatted correctly.

--INFORMATIVE DB 2113-- Default module/procedure cannot be calculated

Condition Identifier: DB\$FS_CANT_CALC_MODULE Code: DB 2113

Description: Debug cannot determine the module and procedure where the variable is defined.

User Action: Press MODIFY and enter the correct module and procedure. Then save that the modified Watch.

--INFORMATIVE DB 2114-- A warning occurred while regenerating source for module {text}. The source for this module may not match the binary. Proceed at your own risk. To correct the error, rerun with a recompiled copy of the module.

Condition Identifier: DBE\$FS_HELP_FOR_REGEN Code: DB 2114

Description: The source code displayed for this module does not correspond to the object code being debugged.

User Action: This is an informative message. If you want both the source and object codes to agree, you must recompile your source.

--INFORMATIVE DB 2115-- Modifying Watch for

Condition Identifier: DBE\$FS_MODIFY_HDR Code: DB 2115

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2116-- OPTION

Condition Identifier: DBE\$FS_OPT_TITLE Code: DB 2116

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2117-- Module

Condition Identifier: DBE\$FS_MODULE_TITLE Code: DB 2117

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2118-- Procedure

Condition Identifier: DBE\$FS_PROC_TITLE Code: DB 2118

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2119-- Kind

Condition Identifier: DBE\$FS_KIND_TITLE Code: DB 2119

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2120-- Stop

Condition Identifier: DBE\$FS_STOP_TITLE Code: DB 2120

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2121-- Log

Condition Identifier: DBE\$FS_LOG_TITLE Code: DB 2121

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2122-- Read

Condition Identifier: DBE\$FS_READ_TITLE Code: DB 2122

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2123-- Write

Condition Identifier: DB\$FS_WRITE_TITLE Code: DB 2123

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2124-- Monitor

Condition Identifier: DB\$FS_MONITOR_TITLE Code: DB 2124

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2125-- Yes

Condition Identifier: DB\$FS_YES_TITLE Code: DB 2125

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2126-- No

Condition Identifier: DB\$FS_NO_TITLE Code: DB 2126

Description: Text displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2127-- Press Next to tab cursor to desired option.

Condition Identifier: DB\$FS_MODIFY_NEXT_TEXT Code: DB 2127

Description: Full-screen message displayed in the Modify window.

User Action: None.

--INFORMATIVE DB 2128-- Enter new value(s).

Condition Identifier: DB\$FS_MODIFY_ENTER_PROMPT Code: DB 2128

Description: Full-screen instructive text.

User Action: Enter the requested information.

--INFORMATIVE DB 2129-- Press Save or Cancel to exit Modify.

Condition Identifier: DB\$FS_EXIT_MODIFY Code: DB 2129

Description: Text for Modify display.

User Action: None.

--INFORMATIVE DB 2130-- Procedure not found

Condition Identifier: DB\$FS_MODIFY_PROC_INVALID Code: DB 2130

Description: The procedure shown on the procedure line of the modify display cannot be found. Perhaps it is misspelled.

User Action: Enter a valid procedure, press CANCEL, or press RESET to get back all values known when modify was entered.

--INFORMATIVE DB 2131-- Wrong combination of values for Kind

Condition Identifier: DB\$FS_MODIFY_KIND_INVALID Code: DB 2131

Description: You cannot combine Kind=Monitor with any other kind in the Modify display.

User Action: Choose between Monitor and the other Kinds specified. Enter a space beside the choice you do not want.

--INFORMATIVE DB 2132-- Module not found

Condition Identifier: DBE\$FS_MODIFY_MODULE_INVALID Code: DB 2132

Description: The module shown on the module line of the modify display cannot be found. Perhaps it is misspelled.

User Action: Enter a valid module, press CANCEL, or press RESET to get back all values known when MODIFY was entered.

--INFORMATIVE DB 2133-- There is no variable to modify

Condition Identifier: DBE\$FS_NO_VAR_TO_MODIFY Code: DB 2133

Description: Debug cannot determine which Watch to modify. The cursor is not on a variable and WATCH was not the most recent function used.

User Action: Place the cursor on the variable in the Watch display to be modified, then press MODIFY.

--INFORMATIVE DB 2134-- Watch brings up the Watch window where you can monitor program identifier values. Output erases the Watch window and brings up the Output window.

Condition Identifier: DBE\$FS_HELP_WATCH_FUNCTION Code: DB 2134

Description: Message is HELP description of the WATCH function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2135-- Modify allows you to change the defaults for Watched program identifiers.

Condition Identifier: DBE\$FS_HELP_MODIFY_FUNCTION Code: DB 2135

Description: Message is HELP description for the MODIFY function displayed in the Help window.

User Action: None.

--INFORMATIVE DB 2136-- Cancel deletes the Watch of the program identifier under the cursor. If there is not a program identifier under the cursor, all watches are cancelled.

Condition Identifier: DBE\$FS_HELP_CANCEL_FUNCTION Code: DB 2136

Description: Full-screen message displaying help information for the CANCEL function.

User Action: None.

--INFORMATIVE DB 2137-- You cannot specify stop=yes when kind is monitor

Condition Identifier: DBE\$FS_BAD_KIND_STOP_COMBO Code: DB 2137

Description: Kind of Monitor updates the watched variable's value each time program execution stops. Thus, specifying Stop=Yes is redundant.

User Action: Either change stop to no in the modify display or specify a different Kind (Read and/or Write).

--INFORMATIVE DB 2138-- You cannot watch the same variable more than once

Condition Identifier: DBE\$FS_DUPLICATE_WATCH Code: DB 2138

Description: You cannot watch the same variable with the same name in the same procedure in the same module.

User Action: None.

--INFORMATIVE DB 2139-- The variable name to be watched is improperly formatted

Condition Identifier: DBE\$FS_BADLY_FORMED_NAME Code: DB 2139

Description: The variable name you entered does not conform to the syntax of the language.

User Action: Correct the syntax.

Further Information: See the description of the DISPLAY_PROGRAM_VALUE command in the Debug usage manual.

--INFORMATIVE DB 2140-- Not enough breaks are left to Watch this variable

Condition Identifier: DBE\$FS_NOT_ENOUGH_BREAKS_WATCH Code: DB 2140

Description: When a Read and/or Write Watch is specified, at least one break is required.

User Action: Do one of the following:

- Use MODIFY to set the Kind to Monitor
- Delete other Watches of Kind Read and/or Write
- Use the DELETE_BREAK command to delete breaks with events of Read_Next_Instruction, Read, Write, Call or Branch.

--INFORMATIVE DB 2141-- A fatal error occurred while regenerating source for module {text}. If you need the source for this module, correct the error and use the zoomin key to regenerate the source again.

Condition Identifier: DBE\$FS_HELP_FOR_REGEN_FATAL Code: DB 2141

Description: The specified module must be initialized.

User Action: Select the Zoom-in display (using the ZMIN function) for the specified module.

--INFORMATIVE DB 2142-- User defined key ({text}):

Condition Identifier: DBE\$FS_HELP_FOR_USER_DEF_KEYS Code: DB 2142

Description: Full-screen help header.

User Action: None.

--INFORMATIVE DB 2143-- '{text}'

Condition Identifier: DBE\$FS_HELP_FOR_COMMAND_STRING Code: DB 2143

Description: Help text for the specified user-defined function.

User Action: None.

--INFORMATIVE DB 2144-- Name error written to output window

Condition Identifier: DBE\$FS_WATCH_NAME_ERROR Code: DB 2144

Description: The name entered for Watch is invalid.

User Action: Press OUTPUT to see the message. Then press WATCH to return to Watch. Move the cursor to the variable name in error. Press CANCEL to remove it.

--INFORMATIVE DB 2145-- Value needed for Kind

Condition Identifier: DBE\$FS_MODIFY_KIND_MISSING Code: DB 2145

Description: In order to define a Watch, at least one Kind must be selected.

User Action: Use RESET to get back the original values for all the items in the Modify display, select a value for Kind, or CANCEL the modify.

--INFORMATIVE DB 2146-- Detailed error message in Output Window

Condition Identifier: DBE\$FS_ERROR_TO_OUTPUT_WINDOW Code: DB 2146

Description: The error message is too long to fit on the message line. The message has been logged in the Output window.

User Action: Use the OUTPUT function to see the message in the Output window.

--FATAL DD 114-- Screen mode cannot be used with batch processing.

Condition Identifier: DDE\$NO_SCREEN_IN_BATCH Code: DD 114

Description: When accessing IM/Control in batch mode with the CONTROL command, you specified MODE=SCREEN. You cannot access screen mode IM/Control in batch mode.

User Action: Specify MODE=COMMAND when accessing IM/Control in batch mode.

--ERROR DD 120-- Parameter {text} is required but was omitted.

Condition Identifier: DDE\$OMITTED_REQUIRED_PARAMETER Code: DD 120

Description: When accessing IM/Control, you omitted a required parameter from the CONTROL command.

User Action: Enter the CONTROL command again, and specify all required parameters.

--INFORMATIVE DD 828-- Errors encountered. The list file, {text}, may have more details.

Condition Identifier: DDE\$ERRORS_ENCOUNTERED Code: DD 828

Description: A fatal or catastrophic error occurred, causing IM/Control to stop executing.

User Action: See your output list; it lists the error that caused IM/Control to stop executing.

--WARNING DD 830-- Unexpected end of information encountered on input file. Missing QUIT command or command missing terminator character.

Condition Identifier: DDE\$PREMATURE_EOI Code: DD 830

Description: Your input file is either missing a QUIT command or contains an IM/Control command that is not followed by a semicolon.

User Action: Edit your input file by either inserting a QUIT command or placing a semicolon after each IM/Control command.

--ERROR DE 0-- {text}.

Condition Identifier: DEE\$SCL_PROCEDURE_ERROR Code: DE 0

Description: An SCL procedure/command generated and executed by the Desktop Environment returned an error.

User Action: See the job log for a description of the SCL error.

--WARNING DE 1-- {text}.

Condition Identifier: DEE\$SCL_PROCEDURE_WARNING Code: DE 1

Description: An SCL procedure/command generated and executed by the Desktop Environment returned a warning error.

User Action: See the job log for a description of the SCL warning error.

--ERROR DE 5-- Internal error detected: {text}.

Condition Identifier: DEE\$INTERNAL_ERROR Code: DE 5

Description: An internal Desktop Environment error has occurred.

User Action: See job log for details. Contact your site analyst.

--ERROR DE 6-- Premature end of information encountered on file {file}.

Condition Identifier: DEE\$PREMATURE_END_OF_FILE Code: DE 6

Description: A premature end of information was found for a file. See job log for details.

User Action: Contact your site analyst.

--ERROR DE 7-- {text} detected {text}.

Condition Identifier: DEE\$ERROR_ABORT Code: DE 7

Description: An internal error caused the Desktop Environment to abort.

User Action: Restart the Desktop Environment. See the job log for details.

--ERROR DE 8-- Received unexpected command {text} from workstation.

Condition Identifier: DEE\$UNEXPECTED_WS_COMMAND Code: DE 8

Description: An internal Desktop Environment error has occurred. Received an unexpected command from the workspace.

User Action: Restart the Desktop Environment. Contact your site analyst.

--ERROR DE 9-- "{text}" is too long to be a valid NOS/VE name.

Condition Identifier: DEE\$USER_NAME_TOO_LONG Code: DE 9

Description: A user name specified is too long to be a NOS/VE name.

User Action: Shorten the specified name to 31 characters or less.

--ERROR DE 10-- "{text}" contains a character that is not allowed in NOS/VE names.

Condition Identifier: DEE\$USER_NAME_BAD_CHARACTER Code: DE 10

Description: A user-entered character string contains characters illegal in a NOS/VE name.

User Action: Ensure names entered are valid NOS/VE names. Characters can be alphabetic or numeric.
However, SCL names cannot start with a numeric character.

--ERROR DE 11-- "{text}" is not a valid NOS/VE name because it starts with a number.

Condition Identifier: DEE\$USER_NAME_NUMERIC Code: DE 11

Description: The user name entered is not a valid NOS/VE name because it starts with a number.

User Action: Ensure entered names for NOS/VE use do not begin with numbers.

--ERROR DE 12-- {text} and {text} can not be specified together for parameter {text}.

Condition Identifier: DEE\$PARAMETER_VALUE_CONFLICT Code: DE 12

Description: The two values specified for the given parameter are mutually exclusive.

User Action: Specify only one of the values for the parameter.

--WARNING DE 13-- Unable to communicate with the FILER because of missing {text} queue.

Condition Identifier: DEE\$NO_FILER_MESSAGE_QUEUE Code: DE 13

Description: An internal Desktop Environment error has occurred. Unable to communicate with the Filer because of missing queue.

User Action: Contact your site analyst.

--ERROR DE 14-- Received unexpected event ORD {text} from Delivery System.

Condition Identifier: DEE\$UNEXPECTED_WS_EVENT Code: DE 14

Description: An internal Desktop Environment error has occurred. Unexpected workstation event.

User Action: Contact your site analyst.

--ERROR DE 15-- Received invalid filer request: {text}.

Condition Identifier: DEE\$INVALID_FILER_REQUEST Code: DE 15

Description: An internal Desktop Environment error has occurred. Invalid Filer request.

User Action: Contact your site analyst.

--ERROR DE 16-- Unexpected value for parameter {text}.

Condition Identifier: DEE\$UNEXPECTED_VALUE_FOR Code: DE 16

Description: The wrong kind of value was specified for the given parameter.

User Action: Specify the correct kind of value.

--ERROR DE 17-- Local queue fap error: {text}.

Condition Identifier: DEE\$LOCAL_QUEUE_FAP_MESS_ERROR Code: DE 17

Description: An internal Desktop Environment error has occurred. Local queue file access procedure error.

User Action: Contact your site analyst.

--ERROR DE 18-- Unable to allocate message buffer in heap.

Condition Identifier: DEE\$CANT_ALLOCATE_MESSAGE_SPACE Code: DE 18

Description: An internal Desktop Environment error has occurred. Unable to allocate message buffer in heap.

User Action: Contact your site analyst.

--ERROR DE 19-- CDNA title pattern too long.

Condition Identifier: DEE\$TITLE_TOO_LONG Code: DE 19

Description: An internal Desktop Environment error has occurred. CDNA title pattern is too long.

User Action: Contact your site analyst.

--ERROR DE 20-- VTP change attribute rejected.

Condition Identifier: DEE\$VTP_CHANGE_ATTR_REJECT Code: DE 20

Description: An internal Desktop Environment error has occurred. VTP change attribute rejected.

User Action: Contact your site analyst.

--ERROR DE 21-- IPG connection broken.

Condition Identifier: DEE\$IPG_CONNECTION_BROKEN Code: DE 21

Description: Internal DE error has occurred. Unable to establish connection with IPG server.

User Action: Contact your site analyst.

--ERROR DE 22-- Received unexpected vtp message during data transmission.

Condition Identifier: DEE\$UNEXPECTED_VTP_MESSAGE Code: DE 22

Description: An internal Desktop Environment error occurred. Received unexpected vtp message during data transmission.

User Action: Contact your site analyst.

--ERROR DE 23-- Unable to establish connection with IPG server.

Condition Identifier: DEE\$CANT_CONNECT_WITH_IPG Code: DE 23

Description: An internal Desktop Environment error has occurred. Unable to establish connection with IPG server.

User Action: Contact your site analyst.

--ERROR DE 24-- VTP query attributes rejected.

Condition Identifier: DEE\$VTP_QUERY_ATTR_REJECT Code: DE 24

Description: An internal Desktop Environment error has occurred. VTP query attributes rejected.

User Action: Contact your site analyst.

--ERROR DE 25-- The Desktop has lost communications with the workstation.

Condition Identifier: DEE\$LOST_COMMUNICATION_WITH_WS Code: DE 25

Description: The Desktop has lost communications with the workstation.

User Action: Restart Desktop VE. See job log for details. Contact your site analyst.

--ERROR DE 26-- Unable to start utility {text}.

Condition Identifier: DEE\$UNABLE_TO_START_UTILITY Code: DE 26

Description: The desktop application could not start-up the specified partner application.

User Action: See your job log for details and contact your site analyst.

--ERROR DE 27-- Utility {text} terminated by dep\$stop_async_utility.

Condition Identifier: DEE\$UNABLE_TO_STOP_UTILITY Code: DE 27

Description: The desktop application could not stop the specified partner application.

User Action: See your job log for details and contact your site analyst.

--ERROR DE 28-- The utility {text} has terminated unexpectedly.

Condition Identifier: DEE\$UNEXPECTED_UTL_TERMINATION Code: DE 28

Description: The desktop application's specified by the partner application terminated unexpectedly.

User Action: See your job log and contact your site analyst.

--ERROR DE 1000-- {text}.

Condition Identifier: DEE\$ERROR Code: DE 1000

Description: The compiler detected a compilation error.

User Action: Correct the error and recompile.

--ERROR DE 1001-- COMPILE_CONTEXT detected {text} errors.

Condition Identifier: DEE\$CONTEXT_COMPILE_ERRORS Code: DE 1001

Description: A number of errors have occurred during the compilation which must be corrected.

User Action: Find the errors in the listing file, correct, and recompile.

--WARNING DE 1002-- COMPILE_CONTEXT detected {text} warnings.

Condition Identifier: DEE\$CONTEXT_COMPILE_WARNINGS Code: DE 1002

Description: A number of warnings have occurred during the compilation which may or may not have to be corrected.

User Action: Find the warnings in the listing, correct if necessary, and recompile.

--ERROR DE 1003-- Name {text} referenced but not defined.

Condition Identifier: DEE\$UNDEFINED_NAME Code: DE 1003

Description: There is no item or structure defined in the module for the specified name.

User Action: Define an item or structure for the specified name, and recompile.

--WARNING DE 1004-- Name {text} referenced but not defined.

Condition Identifier: DEE\$UNDEFINED_NAME_WARNING Code: DE 1004

Description: There is no item or structure defined for the specified name.

User Action: Determine if an item needs to be defined for the specified name. If so, define the item and recompile.

--WARNING DE 1007-- {text} is no longer supported.

Condition Identifier: DEE\$NO_LONGER_SUPPORTED Code: DE 1007

Description: The specified command or parameter is no longer supported by the compiler.

User Action: Delete the command or parameter from the module and recompile.

--WARNING DE 1008-- {text}.

Condition Identifier: DEE\$WARNING Code: DE 1008

Description: The compiler detected a compilation warning.

User Action: Determine if the warning should be corrected. If so, correct and recompile.

--ERROR DE 2000-- The NOS/VE mail utility terminated unexpectedly. See job log for error.

Condition Identifier: DEE\$UNEXPECTED_MAIL_TERMINATION Code: DE 2000

Description: The NOS/VE mail utility terminated unexpectedly.

User Action: See job log for details.

--ERROR DE 2001-- Unexpected output from DISPLAY_LETTERS ({text}). See job log for error.

Condition Identifier: DEE\$UNEXPECTED_DISL_OUTPUT Code: DE 2001

Description: Unexpected output from DISPLAY_LETTERS.

User Action: See job log for details.

--ERROR DE 2002-- {text} not specified on SEND LETTER.

Condition Identifier: DEE\$MISSING_XXXXX_ON_SENL Code: DE 2002

Description: Parameter not specified on SEND LETTER.

User Action: Ensure all required parameters are specified.

--ERROR DE 2003-- RECHECKing the mailbox has made it impossible to do a FORWARD or REPLY TO with this outgoing letter. Save the body of this letter by copying it into the clipboard, discard the outgoing letter, and try again.

Condition Identifier: DEE\$LOST_LETTER_TO_FWD_OR_REPLY Code: DE 2003

Description: Rechecking the mailbox has made it impossible to do a FORWARD or REPLY TO with the current letter.

User Action: Save the body of the letter by copying it into the clipboard. Discard the outgoing letter and try again.

--ERROR DE 2004-- There is already one instance of MAIL running.

Condition Identifier: DEE\$MAIL_IS_ALREADY_RUNNING Code: DE 2004

Description: There is already a NOS/VE mail task running.

User Action: Terminate the previously running mail task.

--ERROR DE 2005-- Unable to start MAIL/VE application.

Condition Identifier: DEE\$UNABLE_TO_START_MAILVE Code: DE 2005

Description: Some NOS/VE or NOS/VE mail error is preventing mail from starting.

User Action: Contact your site analyst.

--ERROR DE 2006-- Unable to find output from MAIL/VE utility.

Condition Identifier: DEE\$MISSING_MAIL_OUTPUT_FILE Code: DE 2006

Description: The desktop mail application was unable to find the output file from Mail/VE.

User Action: Contact your site analyst.

--ERROR DE 2100-- Too many catalogs and object_libraries encountered by the Desktop.

Condition Identifier: DEE\$RAN_OUT_OF_WINDOWS Code: DE 2100

Description: The cabinets currently on the user's desktop contain more files and catalogs than the Filer can keep track of.

User Action: Remove one or more user cabinets from the screen and use the recheck application from the system toolbox to show all remaining files and catalogs.

--INFORMATIVE DE 2101-- The following unexpected error was encountered by the Desktop ({text}):

Condition Identifier: DEE\$UNEXPECTED_ERROR_REPORTED Code: DE 2101

Description: An internal unexpected desktop error occurred.

User Action: See job log for more detail. Retry the previous action. Contact your site analyst.

--ERROR DE 2102-- Could not find window. Retry selection.

Condition Identifier: DEE\$COULD_NOT_FIND_WINDOW Code: DE 2102

Description: An error in searching internal lists has occurred.

User Action: Use the recheck application from the system toolbox to show all remaining files and catalogs. Retry the previous action and contact a site analyst.

--ERROR DE 2103-- Could not find file. Retry selection.

Condition Identifier: DEE\$COULD_NOT_FIND_FILE Code: DE 2103

Description: Could not find the selected file.

User Action: Use the recheck application from the system toolbox, then retry selecting the file.

--ERROR DE 2104-- Cabinet "{text}" is already on your Desktop.

Condition Identifier: DEE\$DUPLICATE_CABINET Code: DE 2104

Description: The specified cabinet is already present on your desktop.

User Action: None.

--ERROR DE 2105-- User cabinet can only be moved to the "Trash".

Condition Identifier: DEE\$MUST_DISCARD_TO_TRASH Code: DE 2105

Description: User cabinets can only be moved to TRASH.

User Action: Only move user cabinets to TRASH.

--ERROR DE 2106-- Maximum number of cabinets ({text}) are on your Desktop.

Condition Identifier: DEE\$RAN_OUT_OF_PATHS Code: DE 2106

Description: The maximum number of cabinets are on your desktop.

User Action: Drag a cabinet to TRASH to be able to display another cabinet.

--ERROR DE 2107-- Family name not specified for user "{text}".

Condition Identifier: DEE\$MISSING_FAMILY_NAME Code: DE 2107

Description: The family name was not specified for the user.

User Action: Specify the user's family.

--ERROR DE 2108-- User name not specified for family "{text}".

Condition Identifier: DEE\$MISSING_USER_NAME Code: DE 2108

Description: The user name was not specified for this family.

User Action: Specify the user name for the family.

--ERROR DE 2110-- Ran out of space in window "{text}".

Condition Identifier: DEE\$REACHED_RECD_LIMIT Code: DE 2110

Description: The cabinets currently on the user's desktop contain more files and catalogs than the Filer can keep track of.

User Action: Remove one or more user cabinets from the screen, and use the recheck application from the file_applications folder of the system toolbox to show all remaining files and catalogs.

--ERROR DE 2111-- Ran out of space for window "{text}".

Condition Identifier: DEE\$REACHED_WINDOW_LIMIT Code: DE 2111

Description: The cabinets currently on the user's desktop contain more files and catalogs than the Filer can keep track of.

User Action: Remove one or more user cabinets from the screen, and use the recheck application from the file_applications folder of the system toolbox to show all remaining files and catalogs.

--INFORMATIVE DE 2112-- Application requested rescan starting at {file}.

Condition Identifier: DEE\$APP_RESCAN_REQUEST Code: DE 2112

Description: Application requested rescan of catalogs.

User Action: None.

--INFORMATIVE DE 2113-- During application-requested rescan starting at {file}, encountered following error:

Condition Identifier: DEE\$APP_RESCAN_ERROR Code: DE 2113

Description: An error occurred during a rescan of catalogs requested by an application.

User Action: None.

**--INFORMATIVE DE 2114-- The Desktop does not have a file cabinet for
{text}.**

Condition Identifier: DEE\$CABINET_NOT_KNOWN Code: DE 2114

Description: The desktop does not have a file cabinet for this user.

User Action: Check the user name specified for the cabinet.

--INFORMATIVE DE 2115-- Could not find all of path in the Desktop tables.

Condition Identifier: DEE\$NOT_ALL_OF_PATH_FOUND Code: DE 2115

Description: An internal error occurred indicating an error in searching tables.

User Action: Use the recheck application from the system toolbox, then retry the previous action. Contact your site analyst.

**--FATAL DE 2116-- The Desktop cannot continue execution due to the error
listed in the job log ({text}).**

Condition Identifier: DEE\$FATAL_FILER_ERROR Code: DE 2116

Description: The Desktop cannot continue due to the error listed in the job log.

User Action: See the job log for a description of the error. Contact your site analyst.

**--INFORMATIVE DE 2117-- Received unexpected user command while waiting
for dialog response.**

Condition Identifier: DEE\$RECEIVED_UNEXPECTED_COMMAND Code: DE 2117

Description: An internal error indicates a bad sequence of commands.

User Action: Retry the previous action. Contact your site analyst.

**--FATAL DE 2118-- The System Desktop Configuration file "{file}" is bad or
missing. Inform your site analyst. The Desktop Environment cannot continue.**

Condition Identifier: DEE\$BAD_SYSTEM_CONFIG_FILE Code: DE 2118

Description: The System Desktop Configuration file cannot be read or is missing.

User Action: Contact your site analyst.

**--FATAL DE 2119-- Your Desktop Configuration file cannot be read. See your job
log for details. The Desktop Environment cannot continue.**

Condition Identifier: DEE\$BAD_USER_CONFIG_FILE Code: DE 2119

Description: Your Desktop Configuration file cannot be read. The Desktop Environment cannot continue.

User Action: Delete your desktop configuration file DEF\$DESKTOP_CONFIGURATION, then reenter the Desktop Environment to rebuild the configuration file. If this fails, contact your site analyst. Note: User defined application alias lists and definitions will be lost.

**--FATAL DE 2120-- Could not open your Desktop Configuration file. See your job
log for details. The Desktop Environment cannot continue.**

Condition Identifier: DEE\$CANNOT_OPEN_CONFIG_FILE Code: DE 2120

Description: Your Desktop Configuration file could not be opened. The Desktop Environment cannot continue.

User Action: See the job log for details. Check the permission of DEF\$DESKTOP_CONFIGURATION. If necessary, delete your desktop configuration file DEF\$DESKTOP_CONFIGURATION, in \$user. \$desktop_ then reenter the Desktop Environment to rebuild the configuration file. If this fails, contact your site analyst. Note: User defined application alias lists and definitions will be lost.

**--ERROR DE 2121-- Could not copy/move the NULL file -> "{text}". Hold down
the option key while dragging the file to TRASH to delete it.**

Condition Identifier: DEE\$COULD_NOT_COPY_FILE Code: DE 2121

Description: An attempt was made to copy an empty file. It is not possible to copy empty files.

User Action: Do not attempt to copy empty files.

--ERROR DE 2122-- The destination of this copy or move exists. Delete the Copy of ... {text} .

Condition Identifier: DEE\$FILE_TO_COPY_TO_EXISTS Code: DE 2122

Description: The destination of this copy exists.

User Action: Delete the destination file or catalog. In the case of a duplicate command this file will be 'copy of <source file>'.

--ERROR DE 2125-- Move of catalog failed. Some file or files in the catalog are probably busy or you are not permitted access to the destination catalog/cabinet.

Condition Identifier: DEE\$MOVE_CATALOG_FAILED Code: DE 2125

Description: Move of catalog failed. Some file or files in the catalog are probably busy or you are not permitted access to the destination catalog/cabinet.

User Action: Examine the file/catalog permissions of the items to be moved. Change the permissions to include "control permission" to allow the move to be performed.

--ERROR DE 2200-- Command {text} from the workstation was unexpected.

Condition Identifier: DEE\$UNEXPECTED_COMMAND_FROM_WS Code: DE 2200

Description: An internal Desktop Environment error has occurred. Unexpected command received from the workspace.

User Action: Restart the Desktop Environment. Contact your site analyst.

--ERROR DE 4000-- The TOOLBOX folder can't be exited.

Condition Identifier: DEE\$CANT_POP_TOOLBOX_FOLDER Code: DE 4000

Description: The toolbox folder can't be exited in this fashion.

User Action: Use GENERATE_TOOLBOX and QUIT commands to exit and save changes.

--ERROR DE 4001-- {text} already exists in the working folder.

Condition Identifier: DEE\$TOOLBOX_ENTRY_ALREADY_EXIST Code: DE 4001

Description: The entry already exists in the working folder.

User Action: Use a different name for the entry.

--ERROR DE 4002-- {text} is not an entry in the working folder.

Condition Identifier: DEE\$ENTRY_NOT_IN_FOLDER Code: DE 4002

Description: The entry is not in the working folder.

User Action: None.

--ERROR DE 4003-- {text} is not a folder.

Condition Identifier: DEE\$ENTRY_IS_NOT_A_FOLDER Code: DE 4003

Description: Entry was attempted to be used as a folder but is not a folder.

User Action: Select a folder to perform folder related actions.

--ERROR DE 4004-- {text} is not an application.

Condition Identifier: DEE\$ENTRY_IS_NOT_AN_APPLICATION Code: DE 4004

Description: The application parameter is not an application, probably is a folder.

User Action: Specify an application for the application parameter.

--ERROR DE 4005-- The file {file} already exists and it is not a command file.

Condition Identifier: DEE\$FILE_IS_NOT_A_COMMAND_FILE Code: DE 4005

Description: Attempt was made to GENERATE_TOOLBOX to a file in command format but the existing file is not in command (text) format.

User Action: Specify a new file to contain the GENERATE_TOOLBOX commands.

--ERROR DE 4006-- Unable to allocate {text} in the new toolbox file.

Condition Identifier: DEE\$NO_SPACE_IN_NEW_TOOLBOX Code: DE 4006

Description: Unable to allocate space in the new toolbox file. Internal Desktop Environment error has occurred.

User Action: Contact your site analyst.

--ERROR DE 4007-- Premature end of file encountered in {file} while NEXting {text}.

Condition Identifier: DEE\$EOF_IN_EXISTING_TOOLBOX Code: DE 4007

Description: Premature end of file encountered in Toolbox file.

User Action: Regenerate toolbox file. Contact your site analyst.

--ERROR DE 4008-- Invalid toolbox version {text} in {file}.

Condition Identifier: DEE\$INVALID_TOOLBOX_VERSION Code: DE 4008

Description: Invalid toolbox version, probably an old version of a toolbox.

User Action: Ensure the proper version of the toolbox file is present. A new version can be generated with the GENERATE_TOOLBOX command. Contact your site analyst.

--ERROR DE 4009-- Attempted to QUIT without saving the working toolbox.

Condition Identifier: DEE\$QUIT_WITHOUT_SAVING_TOOLBOX Code: DE 4009

Description: Attempted to QUIT without saving the working toolbox.

User Action: Save the working toolbox prior to exiting.

--ERROR DE 4100-- No command string can be specified with the application keyword '{text}'.

Condition Identifier: DEE\$WANT_APP_KEYWORD_ONLY Code: DE 4100

Description: No command string should be specified for this application keyword.

User Action: Clear the command string.

--ERROR DE 4101-- No command string was specified for application '{text}'.

Condition Identifier: DEE\$NEED_COMMAND_STRING Code: DE 4101

Description: No command string was specified for the application.

User Action: Specify a command string for the application.

--ERROR DE 4102-- No changes were specified for '{text}'.

Condition Identifier: DEE\$NO_APPLICATION_CHANGES Code: DE 4102

Description: No changes were specified for the application alias and/or the application command.

User Action: Specify changes for the application alias or application command.

--ERROR DE 4103-- Application keyword '{text}' cannot be used with file content type '{text}'.

Condition Identifier: DEE\$RESTRICTED_USE_OF_APP_KEY Code: DE 4103

Description: The application alias keyword can't be used with this content type.

User Action: Specify a different application alias name.

--ERROR DE 4104-- OPEN is always the default application for file content '{text}'.

Condition Identifier: DEE\$OPEN_ALWAYS_DEFAULT_APP Code: DE 4104

Description: For content types catalog, object library, toolbox, and user cabinets, 'OPEN' is always the default.

User Action: None.

--ERROR DE 4105-- Cannot change to/from a special keyword alias.

Condition Identifier: DEE\$KEY_APP_CANT_BE_CHANGED Code: DE 4105

Description: It is not possible to change to and from a special keyword alias.

User Action: Do not attempt to change special keyword aliases.

--ERROR DE 4106-- Could not create the {text} in the desktop configuration file.

Condition Identifier: DEE\$COULDNT_CREATE_CONFIG_FILE Code: DE 4106

Description: The Desktop Environment configuration file could not be built or rebuilt.

User Action: Delete the Desktop Environment configuration file, then reenter the Desktop Environment to rebuild the configuration file. If this fails, contact your site analyst. Note: User defined application alias lists and definitions will be lost.

--ERROR DE 4107-- Premature end of file encountered in {file} while reading {text}.

Condition Identifier: DEE\$EOF_IN_EXISTING_CONFIG Code: DE 4107

Description: Premature end of file encountered while reading the configuration file.

User Action: Delete your desktop configuration file DEF\$DESKTOP_CONFIGURATION, then restart the Desktop Environment. A new configuration file will be rebuilt. Note: Your application aliases and application list information will be lost.

--ERROR DE 4108-- Invalid configuration version {text} in {file}.

Condition Identifier: DEE\$INVALID_CONFIG_VERSION Code: DE 4108

Description: The version of def\$desktop_configuration in your \$desktop_ve catalog is invalid. Delete def\$desktop_configuration in the \$desktop_ve catalog then restart Desktop/VE.

User Action: Delete your desktop configuration file DEF\$DESKTOP_CONFIGURATION, in \$user.\$desktop_ve then restart the Desktop Environment. A new configuration file will be rebuilt. Note: Your application aliases and application list information will be lost.

--ERROR DE 4109-- The alias '{text}' is already in the application list.

Condition Identifier: DEE\$APP_ALIAS_ALREADY_IN_LIST Code: DE 4109

Description: The application alias is already in the application list.

User Action: Delete the old alias to add a new one with the same name. Use the APPLICATION_ALIASES command.

--ERROR DE 4110-- Unable to increase the size of the {text} list.

Condition Identifier: DEE\$CANT_GROW_CONFIG_LIST Code: DE 4110

Description: The maximum number of applications have been added to the application list.

User Action: Delete applications from the application list to be able to add other applications.

--ERROR DE 4111-- The alias '{text}' is not in the application list.

Condition Identifier: DEE\$ALIAS_NOT_IN_LIST Code: DE 4111

Description: The application alias is not in the application list.

User Action: Add the application alias to the application list with the APPLICATION_ALIASES command.

--ERROR DE 4112-- '{text}' is not a recognized choice for content type.

Condition Identifier: DEE\$INVALID_CONTENT_TYPE Code: DE 4112

Description: An invalid content type has been specified.

User Action: Specify a valid NOS/VE, Desktop Environment file content type.

--ERROR DE 4113-- '{text}' is not in the processor list.

Condition Identifier: DEE\$PROCESSOR_NOT_IN_LIST Code: DE 4113

Description: The selected processor is not in the processor list.

User Action: Add the processor to the processor list or select another processor.

--ERROR DE 4114-- Alias '{text}' is already in the processor application list.

Condition Identifier: DEE\$ALIAS_INDEX_ALREADY_IN_LIST Code: DE 4114

Description: Application alias is already in processor application list.

User Action: Please use an alias name not already used or delete it.

--ERROR DE 4115-- Alias '{text}' is not in the processor application list.

Condition Identifier: DEE\$ALIAS_INDEX_NOT_IN_LIST Code: DE 4115

Description: The alias is not in the processor application list.

User Action: Add the alias to the application processor list using the APPLICATION_LISTS command.

--ERROR DE 4116-- Unable to allocate space for a new '{text}' list.

Condition Identifier: DEE\$NO_SPACE_FOR_CONFIG_LIST Code: DE 4116

Description: There is not enough room left in the configuration file to change the list.

User Action: Restart the Desktop Environment. Contact your site analyst.

--ERROR DE 4117-- No application alias was given.

Condition Identifier: DEE\$NEED_APPLICATION_ALIAS Code: DE 4117

Description: No application alias was given.

User Action: Specify an application alias.

--ERROR DE 4118-- No processor name was given.

Condition Identifier: DEE\$NEED_PROCESSOR_NAME Code: DE 4118

Description: No processor name was given.

User Action: Specify a processor name.

--ERROR DE 4119-- No processors can be associated with this content type.

Condition Identifier: DEE\$THIS_ICON_HAS_NO_PROCESSORS Code: DE 4119

Description: No processor can be specified with catalog and cabinet content type icons.

User Action: Specify UNKNOWN as the processor for these icon types (catalog, cabinet).

--ERROR DE 4200-- Parameter list is larger or more complex than can be made into a parameter dialog.

Condition Identifier: DEE\$PARAM_LIMIT_EXCEEDED Code: DE 4200

Description: Parameter list is larger or more complex than can be made into a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4201-- Parameter {text} has a type that cannot be included in a parameter dialog.

Condition Identifier: DEE\$UNKNOWN_PARAM_TYPE Code: DE 4201

Description: A parameter has a type that cannot be included in a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4202-- The value of parameter {text} cannot be entered in a parameter dialog.

Condition Identifier: DEE\$UNKNOWN_VALUE_TYPE Code: DE 4202

Description: The value of a parameter cannot be entered in a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4203-- An illegal SCL data value was encountered in a parameter dialog.

Condition Identifier: DEE\$ILFORMED_SCL_VALUE Code: DE 4203

Description: The parameter dialog contains an SCL value that can not be displayed.

User Action: Currently a product limitation.

--ERROR DE 4204-- Parameter {text} is a union of more than two types, and cannot be included in a parameter dialog.

Condition Identifier: DEE\$UNION_OF_MORE_THAN_TWO Code: DE 4204

Description: A parameter is a union of more than two types, and cannot be included in a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4205-- Parameter {text} does not involve a keyword type, and cannot be used in a parameter dialog.

Condition Identifier: DEE\$UNION_WITHOUT_KEYWORD Code: DE 4205

Description: A parameter does not involve a keyword type, and cannot be used in a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4206-- The text representation of parameter {text} is too large to be entered in a parameter dialog.

Condition Identifier: DEE\$VALUE_TEXT_TOO_BIG Code: DE 4206

Description: The text representation of the parameter is too large to be entered in a parameter dialog.

User Action: Currently a product limitation.

--ERROR DE 4207-- Parameter {text} is required but was omitted.

Condition Identifier: DEE\$REQUIRED_PARM_NOT_SPECIFIED Code: DE 4207

Description: A parameter is required but was omitted.

User Action: Specify the required parameters.

--ERROR DE 5000-- Startup application aborted with message "{text}"

Condition Identifier: DEE\$STARTUP_APP_ABORT Code: DE 5000

Description: An internal Desktop Environment error. The Desktop Environment startup application aborted.

User Action: Contact your site analyst.

--ERROR DE 5001-- Unable to establish connection with the delivery system in {text}.

Condition Identifier: DEE\$CANT_CONNECT_TO_DELIV_SYS Code: DE 5001

Description: An internal Desktop Environment error. Unable to establish connection with the delivery system.

User Action: Contact your site analyst.

--ERROR DE 5002-- Unexpected message from the delivery system in {text}.

Condition Identifier: DEE\$UNEXPECTED_DELIV_SYS_MESSGE Code: DE 5002

Description: An internal Desktop Environment error occurred. Unexpected message from delivery system.

User Action: Restart the Desktop Environment. Contact your site analyst.

--WARNING DE 5003-- Unknown request received by delivery system in application queue.

Condition Identifier: DEE\$UNKNOWN_APP_REQUEST Code: DE 5003

Description: An internal Desktop Environment error has occurred. Unknown request received by delivery system in application queue.

User Action: Contact your site analyst.

--ERROR DE 5004-- Workspace name {text} out of range in call to Delivery System.

Condition Identifier: DEE\$ILLEGAL_WS_NAME Code: DE 5004

Description: An internal Desktop Environment error has occurred. Workspace name out of range.

User Action: Restart the Desktop Environment. Contact your site analyst.

--ERROR DE 5005-- Message manager encountered an incomplete message from the workstation while searching for {text}.

Condition Identifier: DEE\$INCOMPLETE_DS_MESSAGE Code: DE 5005

Description: An internal Desktop Environment error has occurred. Message manager has encountered incomplete message from the workstation.

User Action: Contact your site analyst.

--ERROR DE 5006-- Message manager encountered an unexpected message kind {text} from the workstation.

Condition Identifier: DEE\$UNEXPECTED_DS_MESSAGE_KIND Code: DE 5006

Description: An internal Desktop Environment error has occurred. Message manager encountered an unexpected message kind from the workstation.

User Action: Contact your site analyst.

--ERROR DE 5007-- Message manager encountered a {text} for {text} which must be less than the context partition.

Condition Identifier: DEE\$NAME_GTE_CONTEXT_PARTITION Code: DE 5007

Description: An internal Desktop Environment error has occurred. Message manager encountered name which is greater than the workspace partition.

User Action: Contact your site analyst.

--ERROR DE 5008-- No message found to dequeue.

Condition Identifier: DEE\$NO_MESSAGE_IN_SLOT Code: DE 5008

Description: An internal Desktop Environment error has occurred. No message to dequeue.

User Action: Contact your site analyst.

--ERROR DE 5009-- Too many simultaneous applications.

Condition Identifier: DEE\$TOO_MANY_APPLICATIONS Code: DE 5009

Description: An internal Desktop Environment error has occurred. Too many simultaneous applications are active.

User Action: Wait for an application to terminate or cancel an active application prior to starting another application.

--ERROR DE 5010-- Application command too long.');

Condition Identifier: DEE\$COMMAND_LINE_TOO_LONG Code: DE 5010

Description: The application command is too long.

User Action: Shorten the application command. If this does not work, contact your site analyst.

--ERROR DE 5011-- A application interface request was made to unknown application {text}.

Condition Identifier: DEE\$REQUEST_MADE_TO_UNKNOWN_APP Code: DE 5011

Description: An application request was made to an application that terminated.

User Action: Contact your site analyst.

--ERROR DE 5012-- Find for {text} was issued on a closed work space.

Condition Identifier: DEE\$FIND_ON_CLOSED_WORK_SPACE Code: DE 5012

Description: The workstation required the specified structure, but the host application has closed the workspace.

User Action: Determine why the host application has prematurely closed the workspace and correct the problem.

--ERROR DE 5013-- Find for {text} was issued for item not on the workspace.

Condition Identifier: DEE\$FIND_FOR_UNKNOWN_STRUCT Code: DE 5013

Description: The workstation required the specified structure, but it is not in the application's workspace.

User Action: Verify that the workstation is requesting the correct structure, and determine why it is not in the workspace.

--ERROR DE 5014-- Unable to allocate {text} in workspace.

Condition Identifier: DEE\$WORKSPACE_OVERFLOW Code: DE 5014

Description: A desktop application has put too many items into its workspace.

User Action: t your site analyst._TEXT

--ERROR DE 6000-- Unable to load startup application {text}.

Condition Identifier: DEE\$UNABLE_TO_LOAD_STARTUP_APPL Code: DE 6000

Description: The simulator was unable to load the specified application.

User Action: Consult the load map for the error and correct.

--ERROR DE 6001-- QUIT terminated the simulator.

Condition Identifier: DEE\$QUIT_TERMINATED_SIMULATOR Code: DE 6001

Description: The simulator terminated the application in response to a QUIT command.

User Action: None.

--ERROR DE 6002-- Item {text} not found on workspace.

Condition Identifier: DEE\$ITEM_NOT_FOUND_ON_WORKSPACE Code: DE 6002

Description: Attempted to display the specified item, but it is not on the workspace.

User Action: Verify number of the item to display is correct, and make sure the host has put it in the workspace.

--ERROR DE 6003-- {text}.

Condition Identifier: DEE\$ITEM_IS_NOT_A_XXX Code: DE 6003

Description: The specified item's type does not match the type of display command.

User Action: Use the proper display command to display the item.

--ERROR DE 6004-- {text}.

Condition Identifier: DEE\$INDEX_GTR_NUMBER_COMPONENTS Code: DE 6004

Description: The index specified is greater than the number of items in the structure.

User Action: Issue the command using a valid index.

--ERROR DE 6500-- Unexpected or unknown protocol data unit identifier.

Condition Identifier: DEE\$PROTOCOL_PROCEDURE_ERROR Code: DE 6500

Description: An internal communications protocol error has occurred.

User Action: Retry the file transfer again. Communication line or hardware problems may be present. Contact your site analyst if retries are not successful.

--ERROR DE 6501-- File transfer unsuccessful, C-U-ABORT received.

Condition Identifier: DEE\$C_U_ABORT_REQUEST Code: DE 6501

Description: The user has cancelled the File transfer.

User Action: This is an informative message only.

--ERROR DE 6502-- File transfer unsuccessful, C-P-ABORT received.

Condition Identifier: DEE\$C_P_ABORT_REQUEST Code: DE 6502

Description: The workstation has aborted the file transfer.

User Action: Correct any errors reported by the workstation and retry the file transfer.

--ERROR DE 6503-- C-SELECT request received without filename parameter.

Condition Identifier: DEE\$SELECT_REQUIRES_FILENAME Code: DE 6503

Description: An internal hardware or protocol error has occurred.

User Action: Retry the file transfer again. Communication line or hardware problems may be present.
Contact your site analyst if retries are not successful.**--ERROR DE 6504-- C-CREATE request received without filename parameter.**

Condition Identifier: DEE\$CREATE_REQUIRES_FILENAME Code: DE 6504

Description: An internal hardware or protocol error has occurred.

User Action: Retry the file transfer again. Communication line or hardware problems may be present.
Contact your site analyst if retries are not successful.**--ERROR DE 6505-- Unknown diagnostic identifier.**

Condition Identifier: DEE\$UNKNOWN_DIAGNOSTIC_ID Code: DE 6505

Description: An unknown diagnostic identifier has been sent to the host by the workstation.

User Action: Retry the file transfer again. Communication line or hardware problems may be present.
Contact your site analyst if retries are not successful.**--ERROR DE 6506-- File transfer unsuccessful, connection terminated.**

Condition Identifier: DEE\$FTS_CONNECTION_BROKEN Code: DE 6506

Description: The microcomputer has broken the connection with the host.

User Action: Retry the file transfer again. Communication line or hardware problems may be present.
Contact your site analyst if retries are not successful.**--ERROR DF 1-- You must be the system administrator to use the {text} command {text}**

Condition Identifier: DFE\$NOT_SYSTEM_ADMINISTRATOR Code: DF 1

Description: Execution of MANFS subcommands must be performed at the system console.

User Action: None.

--ERROR DF 2-- STORNET memory base address must be expressed in multiples of 1000 octal.

Condition Identifier: DFE\$INVALID_BASE_ADDRESS Code: DF 2

Description: The STORNET base address for use by File Server must be specified in multiples of 1000 octal words.

User Action: The command must be re-executed.

--ERROR DF 3-- File Server requires that the system page size be at least 4096 bytes.

Condition Identifier: DFE\$PAGE_SIZE_TOO_SMALL Code: DF 3

Description: File Server does not support a system page size of less than 4096 bytes.

User Action: The system page size must be increased for File Server.

--ERROR DF 5-- The total space required {text} plus the BASE_ADDRESS of {text} exceeds the total memory of the STORNET {text}.

Condition Identifier: DFE\$STORNET_MEMORY_SURPASSED Code: DF 5

Description: The total STORNET memory space required by File Server (xxxx) is determined by the values specified for the number_of_mainframes, divisions_per_mainframe, and data_transfer_size parameters. The total STORNET memory space allocated to File Server is determined by the values specified for the base_address (bbbb), memory_size (aaaa), half_ecs_switch and maintenance_buffer_size parameters.

User Action: Verify the parameters specified for NUMBER_OF_MAINFRAMES, DIVISIONS_PER_MAINFRAME, DATA_TRANSFER_SIZE, and MEMORY_SIZE parameters.

--ERROR DF 6-- The specified model {text} for mainframe_id {text} is not recognized as a valid CYBER 180 model.

Condition Identifier: DFE\$UNKNOWN_MODEL_NUMBER Code: DF 6

Description: The model number +P1 specified in the mainframe id +P2 is not recognized as a valid CYBER 180 model.

User Action: The command must be re-issued with the correct model number.

--ERROR DF 7-- The total size {text} of queue entries must fit in a page of size {text}.

Condition Identifier: DFE\$TOO_MANY_QUEUE_ENTRIES Code: DF 7

Description: The total number of queue entries is constrained by the page size of the system. xxxx is the combined value of NUMBER_OF_MONITOR_QUEUE_ENTRIES and NUMBER_OF_TASK_QUEUE_ENTRIES parameters in the command, bbbb is the page size of the system on which the command was executed.

User Action: Specify lower values for the NUMBER_OF_MONITOR_QUEUE_ENTRIES or NUMBER_OF_TASK_QUEUE_ENTRIES parameters. For a system with a page size of 4096 (1000(16)) the total of these parameters may only be 125.

--ERROR DF 8-- The {text}_id_number has already been used to specify a different {text} mainframe.

Condition Identifier: DFE\$ID_NUMBER_IN_USE Code: DF 8

Description: Either the SERVER_ID_NUMBER in the DEFINE_SERVER command or the CLIENT_ID_NUMBER in the DEFINE_CLIENT command has been used in the previously executed DEFINE_SERVER or DEFINE_CLIENT subcommand.

User Action: Correct the id number and resubmit the subcommand.

--ERROR DF 9-- Internal error: Unable to assign queue entry index, because {text}

Condition Identifier: DFE\$UNABLE_TO_ASSIGN_Q_ENTRY Code: DF 9

Description: An internal error has occurred on the client mainframe preventing requests from being sent to the server. xxxx describes the internal error condition present.

User Action: Contact your site analyst.

--ERROR DF 10-- Family {text} is already defined.

Condition Identifier: DFE\$FAMILY_ALREADY_DEFINED Code: DF 10

Description: Family xxxx was specified twice in the command or the DEFINE_FAMILY command included a family that was processed previously.

User Action: None. Use the "VED File_Server" or "VED FS" display to examine the list of families already defined to the particular server.

--ERROR DF 11-- Family {text} was NOT defined. The maximum families is {text}.

Condition Identifier: DFES\$MAXIMUM_FAMILIES_CONFIGURED Code: DF 11

Description: Family xxxx was not defined by the subcommand because the total number of families (yy) that can be processed by the File Server was exceeded. yy represents the total number of families served by all of the server mainframes.

User Action: None - no more families can be added to the file system until other families are removed. In order to remove families the File Server must be terminated, the server mainframe must be deleted, and then defined again with the correct number of served families.

--ERROR DF 12-- Define subcommand can only be issued when Client/Server is INACTIVE or TERMINATED.

Condition Identifier: DFES\$PP_ACTIVE_DURING_DEFINE Code: DF 12

Description: DEFINE_SERVER and DEFINE_CLIENT subcommands may only be issued when all servers and clients using the same STORNET connection are in TERMINATED or INACTIVE state.

User Action: J Operator should deactivate all mainframes using the same STORNET connection, issue a DEFINE_SERVER or DEFINE_CLIENT subcommand, and activate the mainframes again.

--ERROR DF 13-- The client mainframe {text} is already configured.

Condition Identifier: DFES\$CLIENT_ALREADY_CONFIGURED Code: DF 13

Description: The named system was already defined as client mainframe.

User Action: Check the CLIENT_MAINFRAME_IDENTIFIER parameter in the subcommand and change its value as necessary.

--ERROR DF 14-- Internal Error: The client job {text} is already registered on the server.

Condition Identifier: DFES\$CLIENT_JOB_REGISTERED Code: DF 14

Description: xxxx is the system-generated job name. A mismatch has occurred between the client and the server mainframes with respect to the client jobs that are using the server.

User Action: Contact your site analyst.

--ERROR DF 15-- Only a maximum of {text} client jobs may be connected to the server.

Condition Identifier: DFES\$MAXIMUM_JOBS_CONNECTED Code: DF 15

Description: Currently the file server only supports running number xx of client jobs.

User Action: Wait and resubmit the request at a later time. Contact site analyst.

--ERROR DF 16-- Internal error: The client_job_id is invalid "{text}".

Condition Identifier: DFES\$BAD_CLIENT_JOB_ID Code: DF 16

Description: Jxxxx is the system-generated job name. A mismatch has occurred between the client and the server mainframes with respect to the client jobs that are using the server.

User Action: Contact your site analyst.

--ERROR DF 17-- Internal error: Unable to release queue entry {text}.

Condition Identifier: DFES\$RELEASE_Q_ENTRY_ERROR Code: DF 17

Description: An internal error occurred on the client mainframe indicating improper use of the queueing mechanisms.

User Action: Contact your site analyst.

--ERROR DF 18-- The specified mainframe_id {text} is NOT of the form \$SYSTEM_MMMM_NNNN, Where MMMM is the model number, and NNNN is the serial number. {text}

Condition Identifier: DFES\$IMPROPER_MAINFRAME_ID Code: DF 18

Description: The MAINFRAME_IDENTIFIER in the specified MANAGE_FILE_SERVER utility subcommand was not in the proper format.

User Action: Reenter the subcommand in the correct MAINFRAME_IDENTIFIER format.

--ERROR DF 19-- Internal error: An unexpected buffer version {text} was received. Expected {text}

Condition Identifier: DFE\$PROTOCOL_ERROR_VERSION Code: DF 19

Description: The message received from the partner mainframe contained erroneous buffer version (xx), File Server expected version yy.

User Action: Contact your site analyst.

--ERROR DF 20-- Internal error: Queue entry sequence out of order ({text}). Expecting transaction {text}, received {text}.

Condition Identifier: DFE\$PROTOCOL_ERROR_SEQUENCE Code: DF 20

Description: The transaction count received in a message (yy) does not match the transaction count kept locally (xx).

User Action: Contact your site analyst.

--ERROR DF 21-- File Server's PP driver cannot communicate with STORNET via element_name {text}, iou_name {text}, channel_name {text}. Please verify physical channel connection to STORNET low speed port.

Condition Identifier: DFE\$PATH_TO_STORNET_BROKEN Code: DF 21

Description: The channel or STORNET does not respond to functions. Where nnnn is the ELEMENT_NAME for the STORNET ELEMENT_DEFINITION, iiiii is the IOU_NAME, and cccc is the CHANNEL_NAME.

User Action: Contact a customer engineer. A possible cause of the problem may be that input and output connections to STORNET low speed port are reversed or not connected; STORNET low speed port or channel may be defective.

--ERROR DF 22-- Internal error: Invalid served family index on {text}, pointers index {text}, family_list {text}

Condition Identifier: DFE\$INVALID_SERVED_FAMILY_INDEX Code: DF 22

Description: There is a mismatch between the file system and the File Server on the client side of the configuration. xxxx describes the file system operation being performed, bbbb and aaaa are internal indices provided by the file system.

User Action: Contact your site analyst.

--ERROR DF 23-- Internal error: Driver {text} is unknown.

Condition Identifier: DFE\$UNKNOWN_DRIVER Code: DF 23

Description: The driver specified by the ELEMENT_NAME parameter is unknown to the system.

User Action: The command must be re-executed with the proper value specified for the ELEMENT_NAME parameter.

--ERROR DF 24-- Internal error: Driver {text} is already registered.

Condition Identifier: DFE\$DRIVER_ALREADY_RECORDED Code: DF 24

Description: An internal error has occurred in establishing the list of the File Server drivers on this mainframe. xxxx is the element name of the driver in question.

User Action: Contact your site analyst.

--ERROR DF 25-- Mainframe {text} is not a Client.

Condition Identifier: DFE\$MAINFRAME_NOT_CLIENT Code: DF 25

Description: The mainframe xxxx has not been defined as a client mainframe. It may be a server mainframe or may not have been included in the File Server configuration at all.

User Action: Specify the correct system ID in the command.

--ERROR DF 26-- Mainframe {text} is not a Server.

Condition Identifier: DFE\$MAINFRAME_NOT_SERVER Code: DF 26

Description: Mainframe xxxx is has not been defined as a server mainframe. It may be a client mainframe or may not have been included in the File Server configuration at all.

User Action: Specify the correct system ID in the command.

--ERROR DF 27-- Client mainframe {text} is already Active.

Condition Identifier: DFESCLIENT_ACTIVE Code: DF 27

Description: The server-to-client connection is already attempting to activate.

User Action: Wait for the client mainframe to activate, or if the client mainframe is unavailable, issue a TERMINATE_CLIENT subcommand.

--ERROR DF 28-- Server mainframe {text} is already Active.

Condition Identifier: DFESSERVER_ACTIVE Code: DF 28

Description: The named server xxxx has already been activated. The subcommand issued was redundant.

User Action: None.

--ERROR DF 29-- Client mainframe {text} is not Active.

Condition Identifier: DFESCLIENT_NOT_ACTIVE Code: DF 29

Description: The command was issued for the client mainframe which is inactive. The command is redundant.

User Action: None.

--ERROR DF 30-- Server mainframe {text} is not Active.

Condition Identifier: DFESSERVER_NOT_ACTIVE Code: DF 30

Description: The command Deactivate_Server is redundant. It was issued for the server mainframe which is inactive. The permanent file request must be re-issued.

User Action: None.

--ERROR DF 31-- Client mainframe {text} is not Terminated.

Condition Identifier: DFESCLIENT_NOT_TERMINATED Code: DF 31

Description: The client mainframe definition cannot be deleted unless the client has been terminated. Client xxxx is active, deactivated, or inactive.

User Action: Enter a TERMINATE_CLIENT subcommand before reentering the DELETE_CLIENT subcommand.

--ERROR DF 32-- Server mainframe {text} is not Terminated.

Condition Identifier: DFESSERVER_NOT_TERMINATED Code: DF 32

Description: The server mainframe definition cannot be deleted unless the server has been terminated. Server xxxx is active, deactivated, or inactive.

User Action: Enter a TERMINATE_SERVER subcommand before reentering the DELETE_SERVER subcommand.

--ERROR DF 33-- Server mainframe {text} is already defined.

Condition Identifier: DFESSERVER_ALREADY_DEFINED Code: DF 33

Description: Server xxxx has already been defined. The current command is redundant.

User Action: None.

--ERROR DF 34-- Internal Error: Family {text} is not a served family.

Condition Identifier: DFESFAMILY_NOT_FOUND Code: DF 34

Description: Family xxxx was returned by the server in a response message to the client request to verify a family or a queue. Family xxxx is not known to the client system.

User Action: Contact your site analyst.

--ERROR DF 35-- Client {text} is deactivating, you must wait for deactivation process to complete.

Condition Identifier: DFESCLIENT_DEACTIVATED Code: DF 35

Description: The previously issued DEACTIVATE_CLIENT (on the server mainframe) or DEACTIVATE_SERVER command (on the client mainframe) has not yet completed.

User Action: Wait for the deactivation process to complete.

--ERROR DF 36-- Server {text} is deactivating, you must wait for deactivation process to complete.

Condition Identifier: DFE\$SERVER_DEACTIVATED Code: DF 36

Description: The previously issued DEACTIVATE_SERVER (on the client mainframe) or DEACTIVATE_CLIENT command (on the server mainframe) has not yet completed.

User Action: Resubmit the command after the deactivation process has been completed. The server status can be seen in the VED display on an appropriate served family line or on the server display.

--ERROR DF 37-- Client {text} verification error, {text}.

Condition Identifier: DFE\$CLIENT_VERIFICATION_ERROR Code: DF 37

Description: The client xxxx sent a verification message that does not correspond to the server vvvv configuration parameters. As a result, client xxxx remains in a terminated state and should be deleted (DELETE_CLIENT subcommand) and redefined with correct parameters. bbbb may be one of the following text strings:

Server mainframe name mismatch, received: aaaa, expected: aaaa

Monitor queue entries mismatch, received: aaaa, expected: aaaa

Task queue entries mismatch, received: aaaa, expected: aaaa

STORNET base addresses mismatch: Number of mainframes received = z1 Divisions per mainframe received = z2 STORNET flag base received = z3 STORNET memory base received = z4

Driver queue entries mismatch, received: aaaa, expected: aaaa

Driver destination ID mismatch, ID received: aaaa, expected: aaaa

Driver destination queue index mismatch, received: aaaa, expected: aaaa

Driver source ID mismatch, ID received: aaaa, expected: aaaa

Driver source queue index mismatch, index received: aaaa, expected: aaaa

Server lifetime mismatch, lifetime received = aaaa

Server lifetime/birth date mismatch, lifetime received = aaaa, birthdate received = aaaa

System page size incompatibility: Client page size: aaa < server page size bbbb.

System page size incompatibility: Client page size aaaa <>4K AND <> 8K

Note that this message is generated and displayed on the server mainframe and then transmitted and displayed on the client mainframe.

User Action: The erroneous definition may be that of the client or the server or both; examine the parameters associated with both the DEFINE_CLIENT and DEFINE_SERVER commands. A DELETE_CLIENT or a DELETE_SERVER subcommand is required before DEFINE_CLIENT or DEFINE_SERVER command can be re-issued.

--ERROR DF 38-- Server {text} verification error, {text}.

Condition Identifier: DFE\$SERVER_VERIFICATION_ERROR Code: DF 38

Description: The client mainframe sent a verification message that does not correspond to the server mainframe configuration parameters. The client remains in a terminated state. This message is generated and displayed on the server mainframe and then transmitted and displayed on the client mainframe.

User Action: Delete the incorrect mainframe definition (it may be either the client or the server) and redefine it with the correct parameters.

--ERROR DF 39-- The channel {text} {text} for mainframe {text} is not listed in the element definition for element {text}.

Condition Identifier: DFE\$INVALID_CHANNEL Code: DF 39

Description: This is an error resulting from the fact that interrogation of the configuration management tables finds the STORNET element definition, element name eeee, without an entry for the channel name cccc, iou name iiii.for mainframe mmmm.

User Action: Contact your site analyst.

--ERROR DF 40-- Client {text} is already terminated.

Condition Identifier: DFE\$CLIENT_ALREADY_TERMINATED Code: DF 40

Description: The client named in a TERMINATE_CLIENT subcommand is already terminated.

User Action: None.

--ERROR DF 41-- Internal error: Connection {text} is not an STORNET connection type.

Condition Identifier: DFE\$NOT_STORNET_CONNECTION Code: DF 41

Description: An internal error occurred in the File Server PP management. Connection xxxx is not an STORNET connection type.

User Action: Contact your site analyst.

--ERROR DF 42-- Internal error: Connection {text} must be deactivated before changing connection.

Condition Identifier: DFE\$CONNECTION_NOT_CHANGED Code: DF 42

Description: This is an internal error specifying that a connection must be deactivated before it can be changed.

User Action: Contact your site analyst.

--ERROR DF 43-- Either product number {text} or model number {text} is invalid for STORNET product identification.

Condition Identifier: DFE\$INVALID_STORNET_PRODUCT_ID Code: DF 43

Description: This is an error resulting from the fact that interrogation of the configuration management tables finds the STORNET element with incorrect identification.

User Action: Contact your site analyst.

--ERROR DF 44-- No configured channel found in element definition for element {text}.

Condition Identifier: DFE\$NO_CONFIGURED_CHANNEL Code: DF 44

Description: Interrogation of the configuration management tables has found that the defined STORNET element has no configured IO channels.

User Action: Contact your site analyst.

--ERROR DF 45-- Channel definition for {text} has no configured equipment element {text}.

Condition Identifier: DFE\$NO_CONFIGURED_EQUIPMENT Code: DF 45

Description: Channel definition for xxxx has no equipment element for bbbb.

User Action: An internal error occurred in File Server PP management. Call your site analyst.

--ERROR DF 46-- Server {text} has terminated.

Condition Identifier: DFE\$SERVER_HAS_TERMINATED Code: DF 46

Description: The specified server has terminated during a permanent file request.

User Action: The request will be aborted. Reissue the request when the server is active.

--WARNING DF 47-- Server {text} OS version "{text}" conflicts with Client's OS version "{text}".

Condition Identifier: DFE\$OS_NAME_CONFLICT Code: DF 47

Description: The OS version name on the server mainframe is different from the name on the client mainframe.

User Action: Ensure that the two versions of the operating system are compatible. If they are, no further action is required.

--ERROR DF 48-- Client {text} - lifetime/birthdate mismatch, {text}.

Condition Identifier: DF\$CLIENT_LIFETIME_ERROR Code: DF 48

Description: Lifetime aaaa and/or birthdate bbbb received from the client mainframe does not match the values expected by the server mainframe. Server mainframe will undergo File Server termination processing and will automatically restart itself with the server state set to terminated.

User Action: The operator action on the client depends on the server state: if server state is terminated, the problem will correct itself with the client mainframe retransmitting the queue verification message. The message should be treated as informative in this case. If the server state on the client mainframe is inactive, File Server processing will be terminated on the client mainframe and the operator will have to restart it (with ACTIVATE_SERVER) after investigation of the cause for the server lifetime/birthday discrepancy.

--ERROR DF 49-- Mainframe {text} cannot be deleted while element {text} is still reserved.

Condition Identifier: DF\$ELEMENT_STILL_RESERVED Code: DF 49

Description: A DELETE_CLIENT or DELETE_SERVER subcommand was issued while one or more mainframes using the connection was still active.

User Action: There was more than one server or client with the same connection defined on this mainframe and at least one of them is still active. Deactivate all client and server mainframes using the connection and reenter the delete subcommand.

--ERROR DF 50-- Internal error: The remote procedure call queue entry location is invalid ({text}) on the {text} request.

Condition Identifier: DF\$INVALID_QUEUE_ENTRY_ID Code: DF 50

Description: An internal error occurred. The remote procedure call queue entry is invalid for the specified request.

User Action: Contact your site analyst.

--ERROR DF 51-- Internal error: Family {text} is not a served family.

Condition Identifier: DF\$FAMILY_NOT_SERVED Code: DF 51

Description: An internal error occurred. The family specified in a permanent file request is not a served family.

User Action: Contact your site analyst.

--ERROR DF 52-- Internal error: The server_location is invalid.{text}

Condition Identifier: DF\$INVALID_SERVER_LOCATOR Code: DF 52

Description: An internal error occurred. The server location is invalid.

User Action: Contact your site analyst.

--ERROR DF 53-- internal: The remote procedure call request should be restarted.

Condition Identifier: DF\$RESTART_SERVER_REQUEST Code: DF 53

Description: An internal error occurred. The remote procedure call should be restarted.

User Action: Contact your site analyst.

--ERROR DF 54-- Internal error: An error ({text}) occurred in the file server driver.

Condition Identifier: DF\$DRIVER_ERROR_OCCURRED Code: DF 54

Description: An internal error occurred in the File Server driver.

User Action: Contact your site analyst.

--ERROR DF 55-- Internal error: A checksum error ({text}) occurred in the test code.

Condition Identifier: DFE\$TEST_CHECKSUM_ERROR Code: DF 55

Description: An internal checksum error has occurred in the test code.

User Action: Contact your site analyst.

--ERROR DF 56-- Internal error: Attempting to send more data than has been referenced.

Condition Identifier: DFE\$DATA_LENGTH_ERROR Code: DF 56

Description: An internal error occurred. The request is attempting to send more data than has been referenced.

User Action: Contact your site analyst.

--ERROR DF 57-- A served family cannot be specified when the number of monitor queue entries (specified on the DEFINE_SERVER subcommand) is zero.

Condition Identifier: DFE\$NO_FAMILIES_WHEN_ZERO_NOMQE Code: DF 57

Description: A served family cannot be specified when the number of monitor queue entries (specified on the DEFINE_SERVER subcommand) is zero.

User Action: Change the DEFINE_SERVER command to specify zero families or NUMBER_OF_MONITOR_QUEUE_ENTRIES as non-zero. If the error occurs on the DEFINE_SERVED_FAMILY subcommand, then DEFINE_SERVER has to be changed or the DEFINE_SERVED_FAMILY subcommand must not be issued.

--ERROR DF 58-- CHANGE_CLIENT_ACCESS of an accessed family cannot be processed while client {text} is ACTIVE, ACTIVATING, or DEACTIVATED.

Condition Identifier: DFE\$CLIENT_TOO_ACTIVE_FOR_CHACA Code: DF 58

Description: A CHANGE_CLIENT_ACCESS request of an accessed family cannot be processed while client xxxx is active, activating, or deactivated.

User Action: Wait until the client mainframe is inactive or terminated before executing CHANGE_CLIENT_ACCESS.

--ERROR DF 59-- The total number of of accessible {text} ({text}) would exceed the maximum ({text}) if CHANGE_CLIENT_ACCESS processed the given parameters.

Condition Identifier: DFE\$MAX_FAMILIES_OR_CLIENTS Code: DF 59

Description: The total number of accessible families/clients would exceed the maximum (yyy) if CHANGE_CLIENT_ACCESS processed the given parameters.

User Action: Contact your site analyst.

--ERROR DF 60-- Too many names specified for side_door_port parameter.

Condition Identifier: DFE\$STORNET_SIDE_DOOR_DEF_ERR Code: DF 60

Description: The SIDE_DOOR_PORT parameter the DEFINE_STORNET_CONNECTION subcommand contains is in error. It may have too many names or there may be too many side door ports in the list.

User Action: Reissue the subcommand with a corrected SIDE_DOOR_PORT parameter.

--ERROR DF 61-- IOU name {text} specified in SEND_CHANNEL parameter is invalid.

Condition Identifier: DFE\$SEND_CHANNEL_INVALID_IOU Code: DF 61

Description: The IOU name specified in the SEND_CHANNEL parameter of a DEFINE_CLIENT or DEFINE_SERVER subcommand is invalid.

User Action: Reenter the subcommand with a corrected IOU name. Valid IOU names are IOU0 and IOU1.

--ERROR DF 62-- STORNET element {text} contains too many low_speed_ports.

Condition Identifier: DFES\$TOO_MANY_LOW_SPEED_PORTS Code: DF 62

Description: Interrogation of the configuration management tables in a DEFINE_STORNET_CONNECTION subcommand finds too many low speed ports in the configuration.

User Action: Contact your site analyst.

--ERROR DF 63-- Internal error: Global_file_name on client does not match the global_file_name on server for the remote system_file_id: {text}

Condition Identifier: DFES\$SFID_GFN_MISMATCH Code: DF 63

Description: The global file name on the client mainframe does not match the global file name on the server mainframe for the remote system_file_id.

User Action: Contact your site analyst.

--ERROR DF 64-- Maximum mainframes for this connection has been reached and they are all Servers or all Clients: definition for {text} rejected.

Condition Identifier: DFES\$TOO_MANY_SERVERS_OR_CLIENTS Code: DF 64

Description: The most recently executed DEFINE_CLIENT or DEFINE_SERVER subcommand was rejected because it was defining the eighth mainframe in the configuration and the mainframes defined up to that point are either all clients or all servers. This implies a configuration of nine mainframes which exceeds the File Server limit of eight mainframes.

User Action: Check the DEFINE_CLIENT and DEFINE_SERVER subcommand parameters for accuracy. Correct to the eight-mainframe limit.

--ERROR DF 66-- STORNET element {text} has already been defined.

Condition Identifier: DFES\$STORNET_ALREADY_DEFINED Code: DF 66

Description: The specified STORNET element is already defined.

User Action: None, if the STORNET name is correct. Otherwise, delete the definition and then redefine it with a new DEFINE_STORNET_CONNECTION subcommand.

--ERROR DF 67-- STORNET element {text} has not been defined with Define_Stornet_Connection command.

Condition Identifier: DFES\$STORNET_NOT_DEFINED Code: DF 67

Description: The named STORNET xxxx has been used in either DEFINE_CLIENT, DEFINE_SERVER, or DISPLAY_STORNET_CONNECTION subcommand as an ELEMENT_NAME parameter. It must be defined with a DEFINE_STORNET_CONNECTION subcommand before DEFINE_CLIENT, DEFINE_SERVER or DISPLAY_STORNET_CONNECTION can use it as the ELEMENT_NAME parameter.

User Action: Issue DEFINE_STORNET_CONNECTION subcommand for this STORNET name then reissue a DEFINE_CLIENT or DEFINE_SERVER subcommand. If the ELEMENT_NAME parameter is incorrect, the subcommand needs to be reissued with a corrected STORNET element name.

--ERROR DF 68-- System Family {text} cannot be defined as Served Family.

Condition Identifier: DFES\$SYSTEM_FAMILY_NOT_ALLOWED Code: DF 68

Description: Family xxxx specified in the command is a \$SYSTEM family.

User Action: Remove the \$SYSTEM family from the list and reissue the command.

--ERROR DF 69-- IOU name {text} specified in RECEIVE_CHANNEL parameter is invalid.

Condition Identifier: DFES\$RECEIVE_CHANNEL_INVALID_IOU Code: DF 69

Description: Valid IOU names are IOU0 and IOU1.

User Action: Correct the parameter and reenter the subcommand.

--WARNING DF 70-- Server {text} maximum segment length {text} conflicts with Client's maximum segment length {text}.

Condition Identifier: DFE\$SEGMENT_LENGTH_CONFLICT Code: DF 70

Description: The segment lengths on the server and client mainframes are different.

User Action: Ensure that the segment lengths are intentionally different. If they are, no further action is required.

--ERROR DF 71-- Client(s) and/or Server(s) are still defined with this STORNET ({text}) connection.

Condition Identifier: DFE\$STORNET_HAS_MAINFRAMES Code: DF 71

Description: The STORNET connection cannot be deleted until the definitions of all mainframes in the configuration are deleted.

User Action: Issue DELETE_CLIENT and/or DELETE_SERVER commands for every client or server defined on this mainframe as using xxxx as STORNET connection.

--ERROR DF 72-- Define_Served_Families cannot be issued while Server is Activating.

Condition Identifier: DFE\$SERVER_IS_ACTIVATING Code: DF 72

Description: An attempt was made to execute a DEFINE_SERVED_FAMILIES command while the server mainframe was activating. DEFINE_SERVED_FAMILIES can be executed at any time except when the server is activating.

User Action: Reissue the command after server is active.

--ERROR DF 73-- Server Poll Reply contains incorrect mainframe, expected {text}, received {text}.

Condition Identifier: DFE\$INCORRECT_SERVER_MAINFRAME Code: DF 73

Description: An internal error occurred on the client or server mainframe. The mainframe name in the poll message is incorrect.

User Action: Contact your site analyst.

--ERROR DF 74-- Internal Error: NIL Family list for Queue or Family verification.

Condition Identifier: DFE\$NIL_FAMILY_LIST Code: DF 74

Description: An internal error occurred. The message is NIL family list for queue or family verification.

User Action: Contact your site analyst.

--ERROR DF 75-- The client mainframe is already activating.

Condition Identifier: DFE\$CLIENT_IS_ACTIVATING Code: DF 75

Description: The server-to-client connection is already attempting to activate.

User Action: Wait for the client mainframe to activate or, if the client is unavailable, issue a TERMINATE_CLIENT command.

--ERROR DF 76-- {text}_ID_number of {text} is greater than number_of_mainframes value of {text}.

Condition Identifier: DFE\$ID_NUMBER_EXCEEDS_NOMF Code: DF 76

Description: The CLIENT_ID_NUMBER and SERVER_ID_NUMBER parameters for both DEFINE_CLIENT and DEFINE_SERVER subcommands may not be greater than the value specified as the number of mainframes for the DEFINE_STORNET_CONNECTION subcommand.

User Action: Re-enter the command with an ID number less than or equal to the value specified as the number of mainframes for the DEFINE_STORNET_CONNECTION subcommand. If it is necessary to re-enter the DEFINE_STORNET_CONNECTION subcommand, note that all previously defined client and server mainframes and the STORNET connection must be deleted.

--ERROR DU 1-- If dump_file is not specified, the restart file must contain data. {file} is empty or non-existent.

Condition Identifier: DUE\$RESTART_FILE_EMPTY Code: DU 1

Description: Either a valid dump file or restart file must be specified with the ANALYZE_DUMP command.

User Action: Retry using a valid dump file or restart file.

--ERROR DU 2-- File {file} is empty.

Condition Identifier: DUE\$FILE_EMPTY Code: DU 2

Description: The file specified is an empty file.

User Action: Retry using a valid file name.

--WARNING DU 4-- The dump file, {file}, is not in an expected format. {text}

Condition Identifier: DUE\$DUMP_FILE_FORMAT_ERROR Code: DU 4

Description: The dump file specified is not entirely in a format recognizable by ANALYZE_DUMP. Some dump records may be missing or incorrect.

User Action: Notify your site analyst.

--ERROR DU 5-- Address {text} is not on the dump file.

Condition Identifier: DUE\$INVALID_ADDRESS Code: DU 5

Description: The specified address is not contained in the central memory image on the dump file.

User Action: Retry using a valid address.

--ERROR DU 6-- The requested virtual address translation is not possible: {text}{text} {text}{text} {text}

Condition Identifier: DUE\$ADDRESS_TRANSLATION_ERROR Code: DU 6

Description: The RMA calculated with the specified address is not on the central memory image on the dump file.

User Action: Retry using a valid virtual address.

--WARNING DU 7-- A page fault was encountered{text} {text}{text} {text}

Condition Identifier: DUE\$PAGE_FAULT Code: DU 7

Description: The specified memory is not contained on the central memory image on the dump file.

User Action: Retry using a valid virtual address.

--ERROR DU 8-- {text} is not on the dump file.

Condition Identifier: DUE\$ELEMENT_NOT_AVAILABLE Code: DU 8

Description: The specified element is not contained on the dump file.

User Action: None.

--ERROR DU 9-- {text} for processor {text} is not on the dump file.

Condition Identifier: DUE\$PRO_ELEMENT_NOT_AVAILABLE Code: DU 9

Description: The processor element specified is not contained on the dump file.

User Action: None.

--WARNING DU 10-- The pointer to this frame is invalid{text} {text}{text} {text}

Condition Identifier: DUE\$INVALID_SF_POINTER Code: DU 10

Description: The virtual address specified for this stack frame has an invalid stack frame pointer.

User Action: None.

--ERROR DU 11-- The exchange parameter cannot be specified if address mode is real_memory_address.

Condition Identifier: DUE\$RMA_AND_EXC_SPECIFIED Code: DU 11

Description: The REAL_MEMORY_ADDRESS parameter cannot be specified if the exchange parameter has been specified.

User Action: Retry using the correct addressing format.

--WARNING DU 12-- Not all memory requested was displayed: {text}

Condition Identifier: DUE\$INCOMPLETE_MEMORY_DISPLAY Code: DU 12

Description: The remaining memory is not contained in the dump.

User Action: None.

--ERROR DU 13-- The maximum number of bytes that can be displayed at one time is {text}. You requested {text}.

Condition Identifier: DUE\$MEMORY_DISPLAY_OVERFLOW Code: DU 13

Description: You exceeded the maximum number of bytes that can be displayed.

User Action: Retry using an acceptable number of bytes.

--INFORMATIVE DU 14-- {text} duplicate lines are suppressed.

Condition Identifier: DUE\$SKIPPED_LINES Code: DU 14

Description: All the lines specified are identical and therefore not displayed.

User Action: None.

--INFORMATIVE DU 15-- The selected exchange package represents a process in Cyber 170 mode.

Condition Identifier: DUE\$PROCESSOR_IN_170_MODE Code: DU 15

Description: The exchange package being used is in CYBER 170 mode.

User Action: None.

--WARNING DU 16-- Too many errors were encountered while processing the dump file. {file} may not be a valid dump file.

Condition Identifier: DUE\$ERROR_LIMIT_EXCEEDED Code: DU 16

Description: The error threshold for processing dump files was exceeded while processing the specified dump file. Either the tape cannot be read due to tape errors, or the dump file may be invalid.

User Action: Check to see if the dump file is valid.

--WARNING DU 17-- An unexpected end of information was received on file {file} while processing {text}. {text}

Condition Identifier: DUE\$UNEXPECTED_EOI Code: DU 17

Description: The dump file was prematurely terminated while processing the specified record.

User Action: Be suspect of the information in the record that was prematurely terminated.

--WARNING DU 18-- The processing of the dump file, {file}, is terminated due to read errors. See file \$ERRORS for details. {text}

Condition Identifier: DUE\$DUMP_FILE_IO_ERROR Code: DU 18

Description: The threshold for uncorrected read errors has been exceeded in processing the dump file.

User Action: See file \$ERRORS for details.

--INFORMATIVE DU 19-- Display terminated: {text} {text} {text}

Condition Identifier: DUE\$DISPLAY_TERMINATED Code: DU 19

Description: A terminal interrupt was received and the display terminated.

User Action: None.

--ERROR DU 20-- {file} is not a restart file.

Condition Identifier: DUE\$IMPROPER_FILE_ATTRIBUTES Code: DU 20

Description: The specified file does not have the attributes of an ANALYZE_DUMP restart file.

User Action: Specify a valid restart file.

--ERROR DU 21-- {file} already exists, but its preserved attributes do not match those of a restart file. Either delete the file or choose another.

Condition Identifier: DUE\$OVERWRITING_WRONG_FILE Code: DU 21

Description: The restart file specified has preserved file attributes that prohibit it from being used as a restart file.

User Action: Delete the file, or choose another name for the restart file.

--WARNING DU 22-- Invalid page descriptor{text} {text}{text} {text}

Condition Identifier: DUE\$SOFT_PAGE_FAULT Code: DU 22

Description: The valid bit is not set in the page table for this page descriptor.

User Action: None.

--WARNING DU 23-- The PVA was translated with an invalid segment descriptor table entry.

Condition Identifier: DUE\$INVALID_SEG_TRANS_POSS Code: DU 23

Description: The PVA was translated with an invalid segment descriptor table entry.

User Action: None.

--WARNING DU 24-- Virtual memory from offset {text} to {text} is paged out.

Condition Identifier: DUE\$PAGES_SKIPPED Code: DU 24

Description: The specified address range is partially paged out and was skipped.

User Action: None.

--ERROR DU 25-- A page fault was encountered{text} {text}{text} {text}

Condition Identifier: DUE\$PAGE_FAULT_ERROR_SEVERITY Code: DU 25

Description: A page fault was encountered while trying to process the specified memory address.

User Action: None.

--ERROR DU 26-- {text} was not found.

Condition Identifier: DUE\$SYMBOL_NOT_FOUND Code: DU 26

Description: The specified symbol was not found in the debug table being used.

User Action: None.

--ERROR DU 27-- No debug table is available.

Condition Identifier: DUE\$DEBUG_TABLE_NOT_AVAIL Code: DU 27

Description: Symbolic addresses cannot be processed without a debug table.

User Action: Specify a valid debug table on the ANALYZE_DUMP command.

--ERROR DU 28-- No debug table entry was found for address {text}.

Condition Identifier: DUE\$NO_ENTRY_FOR_ADDRESS Code: DU 28

Description: The address specified does not have a corresponding table entry in the debug table being used.

User Action: None.

--ERROR DU 29-- Register {text} is not defined for element {text}.

Condition Identifier: DUE\$REGISTER_NOT_DEFINED Code: DU 29

Description: The register specified is not defined for the element specified.

User Action: None.

--ERROR DU 30-- The value entered for {text}: {text} is not a valid mask setting.

Condition Identifier: DUE\$INVALID_MASK Code: DU 30

Description: The value entered is not a valid mask setting.

User Action: Use a valid mask setting.

--ERROR DU 31-- {text} are not on the dump file.

Condition Identifier: DUE\$ELEMENTS_NOT_AVAILABLE Code: DU 31

Description: The specified elements are not on the dump file.

User Action: None.

--ERROR DU 32-- {text} for processor {text} are not on the dump file.

Condition Identifier: DUE\$PRO_ELEMENTS_NOT_AVAILABLE Code: DU 32

Description: The specified processor elements are not of the dump file.

User Action: None.

--WARNING DU 33-- This frame has been overwritten.{text}

Condition Identifier: DUE\$INVALID_STACK_FRAME Code: DU 33

Description: The stack frame encountered has been overwritten and is invalid.

User Action: None.

--ERROR DU 34-- Peripheral processor {text} is not on the dump file.

Condition Identifier: DUE\$PP_NOT_AVAILABLE Code: DU 34

Description: The peripheral processor specified is not contained on the dump file.

User Action: None.

--WARNING DU 35-- Not all memory requested was copied: {text}

Condition Identifier: DUE\$INCOMPLETE_MEMORY_COPY Code: DU 35

Description: The remaining memory is not contained on the dump file.

User Action: None.

--WARNING DU 36-- No memory was displayed: {text}

Condition Identifier: DUE\$NO_MEMORY_DISPLAYED Code: DU 36

Description: The specified memory is not contained on the dump file.

User Action: None.

--WARNING DU 37-- No memory was copied: {text}

Condition Identifier: DUE\$NO_MEMORY_COPIED Code: DU 37

Description: The specified memory is not contained on the dump file.

User Action: None.

--ERROR DU 38-- {text} for processor {text} is not on the dump file.

Condition Identifier: DUE\$REGISTER_FILE_NOT_AVAILABLE Code: DU 38

Description: There was no register file for the specified processor contained on the dump file.

User Action: None.

--ERROR DU 39-- The buffer controlware for channel {text} is not on the restart file.

Condition Identifier: DUE\$BC_ELEMENT_NOT_AVAILABLE Code: DU 39

Description: There was no controlware record dumped for the specified channel.

User Action: None.

--ERROR DU 40-- The specified offset is out of range.

Condition Identifier: DUE\$INVALID_OFFSET Code: DU 40

Description: The specified offset is too large.

User Action: Retry using a valid offset.

--ERROR DU 41-- The specified record is not on the dump file.

Condition Identifier: DUE\$RECORD_NOT_AVAILABLE Code: DU 41

Description: The specified record is not on the dump file.

User Action: None.

--ERROR DU 42-- ALL must be used alone for parameter {text}.

Condition Identifier: DUE\$ALL_MUST_BE_USED_ALONE Code: DU 42

Description: The processing_options parameter cannot be set to both 'all' and some other parameter.

User Action: Retry the ANALYZE_DUMP command specifying the PROCESSING_OPTIONS parameter only once.

--ERROR DU 43-- Central memory is not on the restart file.

Condition Identifier: DUE\$CM_NOT_ON_RESTART_FILE Code: DU 43

Description: The PROCESSING_OPTIONS parameter on the ANALYZE_DUMP command was selected as HARDWARE, so the central memory record is not copied to the restart file.

User Action: If you need to analyze the contents of central memory, use ANALYZE_DUMP PROCESSING_OPTIONS=ALL to read the dump file again.

--INFORMATIVE DU 44-- Processing option ignored if dump file not specified.

Condition Identifier: DUE\$PROCESSING_OPTION_IGNORED Code: DU 44

Description: You cannot change processing options once the dump file has been copied to a restart file.

User Action: If you wish to analyze the contents of central memory, use ANALYZE_DUMP PROCESSING_OPTIONS=ALL to read the dump file again. If you wish to perform mostly hardware error analysis, use PROCESSING_OPTIONS=HARDWARE.

--INFORMATIVE ES 5-- All changes undone.

Condition Identifier: ESE\$ALL_CHANGES_UNDONE Code: ES 5

Description: The object being edited has been restored to the state it was in when it was selected for editing.

User Action: None.

--ERROR ES 10-- ALL not allowed for {text}.

Condition Identifier: ESE\$ALL_NOT_ALLOWED Code: ES 10

Description: ALL is not an acceptable value for the named parameter.

User Action: Use another keyword or keyword combination and try again.

--ERROR ES 30-- Allocation failed for {text}.

Condition Identifier: ESE\$ALLOCATION_FAILED Code: ES 30

Description: The system cannot allocate space for the named structure.

User Action: You may need to have your validation limits changed or do something to make the operation you are attempting smaller.

--ERROR ES 40-- Box marking is not supported for this command.

Condition Identifier: ESE\$BOX_NOT_IMPLEMENTED Code: ES 40

Description: Box marking is not currently supported.

User Action: Specify the required boundaries by a different method.

--INFORMATIVE ES 50-- {text} lines.

Condition Identifier: ESE\$COUNT_OF_LINES Code: ES 50

Description: The specified number of lines were counted.

User Action: None.

--ERROR ES 70-- The current object is not a deck.

Condition Identifier: ESE\$CURRENT_OBJECT_NOT_A_DECK Code: ES 70

Description: You tried a function that executes on decks only.

User Action: Specify the appropriate deck and try again.

--WARNING ES 85-- Deck {text} interlocked for {text}.

Condition Identifier: ESE\$DECK_INTERLOCKED Code: ES 85

Description: Self-explanatory.

User Action: None.

--ERROR ES 100-- Deck modification limit exceeded.

Condition Identifier: ESE\$DECK_MOD_LIMIT_EXCEEDED Code: ES 100

Description: The system allows only 16,383 modifications to a specific deck.

User Action: None.

--FATAL ES 200-- Editor already active in current_task.

Condition Identifier: ESE\$EDITOR_ACTIVE_IN_TASK Code: ES 200

Description: An attempt was made to call the editor while the editor is already active in the current task.

User Action: Use the EDIT_FILE subcommand of the editor or, if another instance of the editor is necessary, enclose it in a task/taskend block.

--ERROR ES 230-- Either TEXT or REPEAT_SEARCH = TRUE is required.

Condition Identifier: ESE\$EITHER_TEXT_OR_RS Code: ES 230

Description: One of the parameters TEXT or REPEAT_SEARCH (with a value of TRUE) must be specified.

User Action: You need to give a text parameter or indicate that the previous text parameter given is to be reused (REPEAT_SEARCH=TRUE).

--ERROR ES 260-- Errors processing deck {text}.

Condition Identifier: ESE\$ERRORS_PROCESSING_DECK Code: ES 260

Description: An error has occurred in trying to close the specified deck.

User Action: Enter the editor subcommand DISPLAY_EDITOR_STATUS to determine if the deck is still active. If the deck is active, you must enter either END or END_DECK with the parameter WRITE_DECK=FALSE to end processing for that deck. Before closing the deck, you can save an image of the modified text for use with the GENERATE_SCU_EDIT command by ending the WRITE_FILE editor subcommand.

--ERROR ES 265-- File {file} {text}.

Condition Identifier: ESE\$FILE_HAS_WRONG_ATTRIBUTE Code: ES 265

Description: The attributes of the permanent file indicate that it is not an appropriate object for the editor to manipulate. This means that either the file-content attribute indicates that the file contains an object code, or that the file-structure attribute indicates that the file contains a library.

User Action: Consult a manual for available commands that can display the content of the object-files or libraries.

--WARNING ES 270-- Due to errors, text was written to {text}.

Condition Identifier: ESE\$FILE_WRITTEN_TO_UNIQUE Code: ES 270

Description: Errors were encountered while trying to write the text. The text has been written to the specified file.

User Action: Use COPY_FILE to move the text to an appropriate file.

--ERROR ES 350-- History limit exceeded for line {text}.

Condition Identifier: ESE\$HISTORY_LIMIT_EXCEEDED Code: ES 350

Description: SCU allows you to modify an individual line only 255 times.

User Action: None.

--ERROR ES 370-- Internal error {text}.

Condition Identifier: ESE\$INTERNAL_ERROR Code: ES 370

Description: The system messed up.

User Action: Contact your site analyst.

--ERROR ES 418-- Invalid range of columns {text} to {text}.

Condition Identifier: ESE\$INVALID_RANGE_OF_COLUMNS Code: ES 418

Description: The range specified on the COLUMNS parameter is invalid.

User Action: Check the allowable range for the COLUMNS parameter and try again. Range is expected to be given in ascending order.

--ERROR ES 455-- Invalid sequence number.

Condition Identifier: ESE\$INVALID_SEQUENCE_NUMBER Code: ES 455

Description: You specified something other than an integer for a sequence number.

User Action: Change the invalid character to an appropriate integer and try again.

--ERROR ES 470-- Last deck on library.

Condition Identifier: ESE\$LAST_DECK_ON_LIBRARY Code: ES 470

Description: You tried to access the next deck of the library while in the last deck. In other words, there is no next deck.

User Action: None.

--WARNING ES 485-- Line contains unprintable Ascii control characters.

Condition Identifier: ESE\$LINE_CONTAINS_CONTROLS Code: ES 485

Description: The editor has been entered with the parameter DISPLAY_UNPRINTABLE_CHARACTERS equal to false. Character(s) in the range 0 to 31 (base 10) and 127 (base 10) have been encountered. Each has been mapped to a space.

User Action: If the use ends the edit session with the WRITE_FILE parameter equal to true or does an explicit WRITE_FILE command, this mapping is written to the external file. If the user ends the edit session with the WRITE_FILE parameter equal to false, no changes are made to the external file.

--ERROR ES 500-- Line identifier missing modification name.

Condition Identifier: ESE\$LINE_ID_MISSING_MOD_NAME Code: ES 500

Description: A line you specified has a line identifier without a modification name.

User Action: Add a modification name to the line identifier and try again.

--ERROR ES 510-- Line length exceeded.

Condition Identifier: ESE\$LINE_LENGTH_EXCEEDED Code: ES 510

Description: The editor encountered a line which exceeds the line length limit specified by the SET_LINE_WIDTH subcommand. If no SET_LINE_WIDTH subcommand has been entered, lines must be no greater than 256 characters.

User Action: Shorten the long line and try again.

--ERROR ES 520-- Line limit exceeded.

Condition Identifier: ESE\$LINE_LIMIT_EXCEEDED Code: ES 520

Description: SCU allows a maximum of 262,143 lines per deck.

User Action: Shorten the deck and try again.

--WARNING ES 530-- Line longer than current width.

Condition Identifier: ESE\$LINE_LONGER_WIDTH Code: ES 530

Description: The editor encountered a line which exceeds the line length limit specified by the SET_LINE_WIDTH subcommand. If no SET_LINE_WIDTH subcommand has been entered, lines must be no greater than 256 if editing a file. Another limit can be established for decks on a library with the WIDTH parameter on the CREATE_DECK and CHANGE_DECK commands.

User Action: Shorten the line or increase the line width.

--ERROR ES 540-- Line {text} inactive.

Condition Identifier: ESE\$LINE_NOT_ACTIVE Code: ES 540

Description: The specified line is not active.

User Action: The \$ACTIVE_IDENTIFIER editor function can be used to determine the nearest active line.

--ERROR ES 550-- Line {text} not found.

Condition Identifier: ESE\$LINE_NOT_FOUND Code: ES 550

Description: The editor was unable to locate the specified line.

User Action: None.

--ERROR ES 580-- Line {text} truncated during mapping of unprintable characters.

Condition Identifier: ESE\$LINE_TRUNCATED_MAPPING Code: ES 580

Description: The editor has been entered with the DISPLAY_UNPRINTABLE_CHARACTERS parameter equal to TRUE. Enough characters have been mapped to longer mnemonics in this line to cause it to be greater than 256 characters. The line will be truncated at this point.

User Action: To write this mapping to an external file, end the edit session with the WRITE_FILE parameter equal to TRUE or enter an explicit WRITE_FILE command. If the edit session ends with the WRITE_FILE parameter equal to FALSE, no changes are made to the external file. Lines that map longer than 256 characters must be dealt with outside the editor.

--ERROR ES 590-- {text} not found after {text}.

Condition Identifier: ESE\$L2_NOT_AFTER_L1 Code: ES 590

Description: The values given for a line are out of range.

User Action: Try again with lines in the right order.

--ERROR ES 600-- {text} not found in search back.

Condition Identifier: ESE\$L2_NOT_FOUND_IN_SEARCH_BACK Code: ES 600

Description: When searching backward, the second line of the line range was not found.

User Action: Check the line range and try again.

--ERROR ES 650-- Modification {text} is not state 0.

Condition Identifier: ESE\$MOD_NOT_STATE_ZERO Code: ES 650

Description: Self-explanatory.

User Action: None.

--ERROR ES 670-- No active lines.

Condition Identifier: ESE\$NO_ACTIVE_LINES Code: ES 670

Description: Either the deck was empty or no matching lines were found.

User Action: Examine the point at which you started a search. You may already have been at the end of the decks.

--ERROR ES 720-- No deck active.

Condition Identifier: ESE\$NO_DECK_ACTIVE Code: ES 720

Description: Most editor subcommands require that you specify the deck or file on which the action is to take place as a parameter on the EDIT_DECK or EDIT_FILE command or the editor subcommands EDIT_DECK or SELECT_DECK.

User Action: Specify the deck to be edited by entering EDIT_DECK or SELECT_DECK.

--ERROR ES 730-- No decks on library.

Condition Identifier: ESE\$NO_DECKS_ON_LIBRARY Code: ES 730

Description: The specified library does not contain any decks.

User Action: None.

--ERROR ES 736-- NONE not allowed for {text}.

Condition Identifier: ESE\$NONE_NOT_ALLOWED Code: ES 736

Description: NONE is not allowed as a keyword value for the parameter named in the message for this command.

User Action: None.

--ERROR ES 750-- Object stack full.

Condition Identifier: ESE\$OBJECT_STACK_FULL Code: ES 750

Description: The user has tried to edit too many objects concurrently.

User Action: Close one or more files or decks before trying to edit a new file or deck.

--INFORMATIVE ES 760-- {text} occurrences found.

Condition Identifier: ESE\$OCCURRENCES_FOUND Code: ES 760

Description: The editor found the specified number of occurrences.

User Action: None.

--ERROR ES 780-- Sequence number too large.

Condition Identifier: ESE\$SEQ_NUMBER_TOO_LARGE Code: ES 780

Description: Self-explanatory.

User Action: None.

--ERROR ES 785-- Space is not a valid word character.

Condition Identifier: ESE\$SPACE_INVALID_IN_WORD Code: ES 785

Description: Self-explanatory.

User Action: None.

--ERROR ES 790-- String longer than search margins.

Condition Identifier: ESE\$STRING_LONGER_WINDOW Code: ES 790

Description: The string you specified is wider than the current search margins.

User Action: Specify a shorter string or expand the search margins.

--ERROR ES 795-- Terminal not identified.

Condition Identifier: ESE\$TERMINAL_NOT_IDENTIFIED Code: ES 795

Description: The terminal type must be specified on the SET_SCREEN_OPTIONS or ACTIVATE_SCREEN subcommands or the CHANGE_TERMINAL_ATTRIBUTES command before screen mode can be established.

User Action: Enter either a SET_SCREEN_OPTIONS or ACTIVATE_SCREEN subcommand or a CHANGE_TERMINAL_ATTRIBUTES command, specifying the terminal type.

--ERROR ES 800-- Text {text} not found.

Condition Identifier: ESE\$TEXT_NOT_FOUND Code: ES 800

Description: The editor could not find the specified text.

User Action: None.

--ERROR ES 810-- Only {text} transactions can be logged.

Condition Identifier: ESE\$TRANSACTION_LIMIT_EXCEEDED Code: ES 810

Description: The amount of space allocated to support the UNDO feature has been exceeded.

User Action: Close the current file or deck and then open it again.

--ERROR ES 825-- Unexpected data in line identifier.

Condition Identifier: ESE\$UNEXPECTED_DATA_IN_LINE_ID Code: ES 825

Description: The line id contained something other than what is supposed to be there. Valid line identifiers consist either of a name (keyword) or a modification name followed immediately by a period and a sequence number, an integer, or some expression that will evaluate as one of the former constructs.

User Action: Check the line identifier in question and try again.

--ERROR ES 880-- Unknown modification name {text}.

Condition Identifier: ESE\$UNKNOWN_MOD_NAME Code: ES 880

Description: The modification name you specified is unknown to the system.

User Action: Check the spelling of the name and try again.

--WARNING ES 885-- Warnings processing command {text}. See \$ERRORS file.

Condition Identifier: ESE\$WARNINGS_DURING_COMMAND Code: ES 885

Description: The \$ERRORS file will contain more information. For example, you may have entered a READ_FILE subcommand which introduced lines longer than the current width.

User Action: Check the \$ERRORS file for more information.

--WARNING ES 888-- Warnings processing deck {text}.

Condition Identifier: ESE\$WARNINGS_DURING_DECK Code: ES 888

Description: Self-explanatory.

User Action: None.

--ERROR ES 900-- The parameters NEW_TEXT and UNTIL are mutually exclusive.

Condition Identifier: ESE\$WITH_FILE_UNTIL Code: ES 900

Description: You cannot specify the NEW_TEXT or UNTIL parameters on the same subcommand.

User Action: Check the subcommand syntax and try again.

--WARNING FA 0-- Improper access level for C170 file.

Condition Identifier: FA\$IMPROPER_ACCESS_LEVEL Code: FA 0

Description: This message indicates an internal error.

User Action: Follow site-defined procedure for reporting system problems.

--FATAL FA 10-- Illegal C170 file name.

Condition Identifier: FA\$ILLEGAL_C170_FILE_NAME Code: FA 10

Description: A CYBER 170 file name can be one through seven characters in length, can contain letters (A through Z) or digits (0 through 9), and must begin with a letter.

User Action: Correct the file name and reenter the command.

--WARNING FA 20-- This operation is not supported for C170 files.

Condition Identifier: FAESIMPROPER_FAP_OPERATION Code: FA 20

Description: The operation cannot be performed.

User Action: Use an alternative method to perform the task.

--WARNING FA 30-- CRM warning on C170 file operation: {text}.

Condition Identifier: FA\$CRM_WARNING Code: FA 30

Description: CYBER Record Manager detected an error. See the CRM Basic Access Methods Reference manual (for NOS) for information about the error.

User Action: Change the FILE command or your program, whichever is necessary to correct the indicated error. Rerun your migration task.

--FATAL FA 40-- CRM error on C170 file operation: {text}.

Condition Identifier: FA\$CRM_ERROR Code: FA 40

Description: CYBER Record Manager detected an error. For additional information about the error, see the CRM Basic Access Methods reference manual (for NOS).

User Action: Change the FILE command or your program, whichever is necessary to correct the error. Recompile your program, if you changed it. Rerun your migration task.

--ERROR FA 50-- Specified C170 file organization not supported.

Condition Identifier: FA\$CRM_FO_BAD Code: FA 50

Description: The FILE command for the CYBER 170 file specifies a file organization (FO) that is not supported by FMA. Specify either FO=SQ (sequential) or FO=WA (word addressable). SQ is the default organization. WA is the organization for a FORTRAN file opened with the specification ACCESS='DIRECT'.

User Action: Correct the FILE command for the CYBER 170 file and rerun the your migration task.

--ERROR FA 60-- Specified C170 key type not supported.

Condition Identifier: FA\$CRM_KT_BAD Code: FA 60

Description: The key type specified on the KT parameter of the 170 FILE command is not supported.

User Action: Verify that proper key type was specified.

--FATAL FA 70-- Outstanding error on C170 file.

Condition Identifier: FA\$CRM_FNF_BAD Code: FA 70

Description: Usually, you have attempted to use a CYBER 170 file that had a previous fatal error about which you were issued a message. Occasionally, the file could have had an error condition before being processed by your program.

User Action: For the usual situation, correct the outstanding error as indicated by the previous message and rerun your program. For the occasional situation, determine and correct the outstanding error, and rerun your program.

--WARNING FA 80-- Redundant open of C170 file.

Condition Identifier: FA\$CRM_DUPLICATE_OPEN Code: FA 80

Description: The 170 file is already open.

User Action: No action needed. The redundant open is ignored.

--FATAL FA 90-- Required memory unavailable for C170 file.

Condition Identifier: FA\$CRM_NO_MEMORY Code: FA 90

Description: There is insufficient memory for the file or for all the files required by your job. The following items are possible causes for this problem:

You have specified too large values for the maximum record (MRL) for CYBER 170 files.

Your validation for memory usage could have been decreased.

You are running the job on a smaller mainframe.

Your migration job attaches more files than the original NOS FORTRAN program used.

User Action: Check the possibilities.

Reduce the memory required for the job by reducing MRL in FILE command for CYBER 170 file. (The MRL values, however, must be no less than the actual minimum record lengths of the records.) If this is the solution, enter corrected FILE commands and rerun your migration task.

Request that your validation for memory usage at the site be increased and rerun your migration task.

--ERROR FA 100-- Abnormal termination of interstate command.

Condition Identifier: FA\$EXEIC_ABNORMAL Code: FA 100

Description: The CYBER 170 command specified in the EXECUTE_COMMAND command failed.

User Action: Change the command as indicated in the message, and reenter the command.

--ERROR FA 110-- Interstate command control file malfunction.

Condition Identifier: FA\$EXEIC_CTL_IO Code: FA 110

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 120-- Interstate command procedure file malfunction.

Condition Identifier: FA\$EXEIC_PROC_IO Code: FA 120

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 130-- Specified Internal Code is not valid for C170 file.

Condition Identifier: FA\$CRM_IC_BAD Code: FA 130

Description: The FILE command specifies an internal code (IC) which is not valid for use with FMA. IC=D64 (for 64-character display code) must be specified or used by default.

User Action: Correct the FILE command for the CYBER 170 file and rerun the FMA task.

--ERROR FA 140-- FL or MRL is unspecified or zero for C170 file.

Condition Identifier: FA\$CRM_NO_MRL Code: FA 140

Description: Specify the maximum record length (MRL) for the file in the FILE command. (Specifying FL for field length works the same way.) For sequential file specify a length equal to or larger than MRL. For direct access files, specify the correct MRL.

User Action: Determine MRL for the file. Then specify MRL=length (length in characters) in the FILE command and rerun the FMA task.

--ERROR FA 150-- Attempt to access C170 file when C170 state disconnected.

Condition Identifier: FA\$STATE_DISCONNECTED Code: FA 150

Description: To access a 170 file, first open the interstate connection using the OPEN_170_STATE command.

User Action: Open the interstate connection.

--ERROR FA 160-- Residual detected in ICF record.

Condition Identifier: FA\$ICF_RESIDUAL Code: FA 160

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system errors.

--FATAL FA 165-- Forced termination of command detected.

Condition Identifier: FA\$SUBCOMMAND_TERMINATED Code: FA 165

Description: The user has terminated the task via the TERMINATE_BREAK command.

User Action: Not applicable.

--ERROR FA 170-- Only skipping by records is permitted when migrating a file.

Condition Identifier: FA\$IMPROPER_MIGRATED_SKIP_OP Code: FA 170

Description: This message indicates an internal error.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 180-- File boundary detected before skip count exhausted.

Condition Identifier: FAES\$SKIP_COUNT_BAD Code: FA 180

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 190-- Commands from outer environments cannot be executed.

Condition Identifier: FAES\$ILLEGAL_OUTER_ENVNR_ACCESS Code: FA 190

Description: You attempted to execute a command that is not known in your environment but is allowed in another environment in FORTRAN FMA. For example, after you have executed the OPEN_170_STATE command you must execute CLOSE_ENVIRONMENT (or QUIT) before you can execute the EXECUTE_MIGRATION_TASK command.

User Action: Execute the CLOSE_ENVIRONMENT or QUIT command to close your current environment so that subsequent commands will be executed in a containing environment.

--WARNING FA 200-- The Cyber 170 file {text} was closed before all records were read and migrated.

Condition Identifier: FAES\$INCOMPLETELY_MIGRATED_FILE Code: FA 200

Description: Your program closed the file before reading all the records in the source file for migration.

User Action: If you want to migrate the entire file, change the program so that it reads all the records and rerun your migration task.

If you did not want to migrate the entire file, ignore the diagnostic.

--ERROR FA 210-- Duplicate specification of C180 input file {file}.

Condition Identifier: FAES\$DUPLICATE_180_LFN Code: FA 210

Description: In the EXECUTE_MIGRATION_TASK command, you have associated the same NOS/VE file with more than one CYBER 170 file; only a one-to-one association is allowed. A particular file can be specified only once in the command.

User Action: Correct and reenter the command.

--ERROR FA 220-- A FORTRAN statement is not allowed to access a C180 file that is the target of a C170 file currently open for migration.

Condition Identifier: FAES\$OPEN_TARGET_WHEN_C170_OPEN Code: FA 220

Description: The MIGRATE_FILES parameter designates a CYBER 170 and CYBER 180 file as source and target files, respectively. Your program specified reading the CYBER 170 file, which automatically results in migrating the data to the CYBER 180 file. When undergoing migration, the target CYBER 180 file cannot be referenced in a FORTRAN I/O statement. Your program has violated this rule.

User Action: You could have specified the incorrect file in the MIGRATE_FILES parameter. Check the file names in the parameter and associate the correct files.

Alternately, you could want the processing to be performed on the CYBER 180 file after migrating data to it. Specify a CLOSE statement for the source CYBER 170 file before specifying I/O for the target CYBER 180. Ensure that your program performs the same kind of I/O processing for all access to the CYBER 180 file. Remember that FMA rewinds both the source and target files when they are closed after migration.

--ERROR FA 230-- Cannot access a C170 file whose target C180 file is currently open.

Condition Identifier: FAES\$OPEN_C170_WHEN_TARGET_OPEN Code: FA 230

Description: The MIGRATE_FILES parameter designates a CYBER 170 and CYBER 180 file as source and target files, respectively. The program has been accessing the target file as a normal NOS/VE file. Then the program specified accessing the source CYBER 170 file, which FMA interprets as a request to migrate the source file. FMA cannot perform the request because the target file is already open.

User Action: You have probably specified the incorrect file in the MIGRATE_FILES parameter. Check the file names in the MIGRATE_FILES parameter and associate the correct files. Enter the correct FILE command if you changed the CYBER 170 file and reenter the EXECUTE_MIGRATION_TASK command.

If you wished to write data to the target file before using it in migration, remember that FMA truncates the target file before migration.

--ERROR FA 240-- Duplicate specification of C170 input file {text}.

Condition Identifier: FAE\$DUPLICATE_170_LFN Code: FA 240

Description: You have specified the CYBER 170 input file more than once in the EXECUTE_MIGRATION_TASK command. A particular file can be specified only once.

User Action: Correct and reenter the command.

--ERROR FA 250-- Record truncation detected. Amount requested is less than amount in record.

Condition Identifier: FAE\$RESIDUAL_RECORD Code: FA 250

Description: Your program did an unformatted read of a sequential file and did not read the complete record.

User Action: Usually, correct the READ statement to read the complete record. If your CYBER 170 file has F type (for fixed length) records, your MRL (or FL) length could be too large or your READ statement is incorrect. Correct READ statement or the FILE command. Recompile and your program and rerun the migration task.

--ERROR FA 260-- Invalid BDP format detected in 170 data conversion.

Condition Identifier: FAE\$INVALID_BDP_DATA Code: FA 260

Description: The 170 file contains bad data.

User Action: Correct the file data before migrating the file.

--WARNING FA 270-- Loss of significance detected on file {text} in record {text}.

Condition Identifier: FAE\$LOSS_OF_SIGNIFICANCE Code: FA 270

Description: While the indicated record on the indicated file was being migrated, a numeric value was changed slightly because of differences in word length or internal numeric format between CYBER 170 and CYBER 180. This error sometimes occurs when migrating a 64-bit CYBER 180 integer to a 60-bit CYBER 170 integer.

User Action: No action necessary, but you should check the input and output record to determine the extent of the change.

--WARNING FA 280-- Overflow detected on file {text} in record {text}.

Condition Identifier: FAE\$OVERFLOW Code: FA 280

Description: A value exceeds the maximum value that can be represented. Zeros are stored in the output field.

User Action: No action necessary. The file can still be migrated, but you should check the original file for bad values.

--WARNING FA 290-- Underflow condition detected on file {text} in record {text}.

Condition Identifier: FAE\$UNDERFLOW Code: FA 290

Description: A value is less than the minimum value that can be represented.

User Action: Remove from your program all values that cannot be converted.

--ERROR FA 300-- Indefinite value detected in 170 data conversion.

Condition Identifier: FAE\$INDEFINITE Code: FA 300

Description: An indefinite value appears on the 170 file.

User Action: The file being migrated contains bad data.

--ERROR FA 310-- Infinite value detected in 170 data conversion.

Condition Identifier: FAE\$INFINITE Code: FA 310

Description: An infinite value appears on the 170 file.

User Action: The file being migrated contains bad data.

--ERROR FA 320-- Bad parameters detected in fap\$ call.

Condition Identifier: FAES\$BAD_PARAMETERS Code: FA 320

Description: An internal error has occurred.

User Action: Follow site defined procedures for reporting system or operational problems.

--ERROR FA 330-- Working storage length is less than data unit length.

Condition Identifier: FAES\$WSL_LESS_THAN_UNIT_LENGTH Code: FA 330

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 340-- The migrated record from C170 file {file} exceeds the maximum record length for its target file.

Condition Identifier: FAES\$MIGRATED_RECORD_GT_MAXRL Code: FA 340

Description: A record from the 170 file exceeds the maximum record length.

User Action: Increase the maximum record length specification.

--ERROR FA 350-- An output operation is not allowed when migrating a file.

Condition Identifier: FAES\$OUTPUT_DURING_MIGRATION Code: FA 350

Description: When you are migrating a file, you must not perform an output operation. The file is automatically migrated as it is input.

User Action: Remove the output statements.

--FATAL FA 360-- The C170 permanent file {text} could not be found.

Condition Identifier: FAES\$C170_PERMFILE_NOT_FOUND Code: FA 360

Description: The 170 file that was specified does not exist.

User Action: Verify that the proper file name was specified.

--WARNING FA 370-- The following errors occurred on file {file} in record conversion {text}: {text}{text}{text}.

Condition Identifier: FAES\$EXTENDED_ACCESS_WARNINGS Code: FA 370

Description: Errors occurred. The specific errors are listed after this message.

User Action: Correct the specific errors listed after this message.

--FATAL FA 380-- This system is not running NOS or NOS/BE dual state, so the interstate facility is not available.

Condition Identifier: FAES\$SYSTEM_NOT_DUAL_STATE Code: FA 380

Description: You must be using 180-NOS/VE in dual state mode in order to perform this operation.

User Action: Either switch to a dual state system or delete the operation.

--FATAL FA 390-- Attempt to write a record exceeding maximum record length to a C170 file.

Condition Identifier: FAES\$RECORD_EXCEEDS_MAXRL Code: FA 390

Description: FMA tried to write a record whose length exceeded the value of the MAXIMUM_RECORD_LENGTH attribute.

User Action: Increase the value specified by the MAXIMUM_RECORD_LENGTH parameter on the SET_FILE_ATTRIBUTES command.

--ERROR FA 391-- An input operation is not allowed to immediately follow an output operation on a

Condition Identifier: FA\$READ_AFTER_WRITE Code: FA 391

Description: The user's FORTRAN program has attempted a sequential READ operation immediately after a sequential WRITE operation. Since a sequential WRITE defines end-of-information after the record written, a READ at that position is defined as an error condition.

User Action: Correct program logic.

--ERROR FA 400-- The C170 and C180 filenames on the migration_files parameter cannot be the same name.

Condition Identifier: FA\$ILLEGAL_MF_PARAM_FOR_TAPES Code: FA 400

Description: The C170 and C180 filenames are the first two values of the MIGRATION_FILES parameter of the EXECUTE_MIGRATION_TASK command of FORTRAN FMA. The EXECUTION_MIGRATION_TASK command is in error because the C170 filename refers to a tape file, and C180 filename is the same as the C170 filename. When these filenames are the same, the C180 filename is assumed to be that of a NOS/VE tape file, which the MIGRATION_FILES parameter cannot support. Therefore the filenames must be different.

User Action: Change either the C170 filename or the C180 filename in the MIGRATION_FILES parameter and anywhere else that the filename appears.

--ERROR FA 410-- An undetermined malfunction occurred in the 170 slave processor. The error condition could not be retrieved.

Condition Identifier: FA\$MALFUNC_IN_FASLAVE Code: FA 410

Description: This message indicates an internal error.

User Action: Follow site-defined procedure for reporting system problems.

--ERROR FA 1000-- No file was selected from COLFD.

Condition Identifier: FA\$NO_COLFD_FILE_SELECTED Code: FA 1000

Description: The INPUT parameter of the COLLECT_FILE_DESCRIPTION command was not properly specified.

User Action: Specify a file using the INPUT parameter.

--ERROR FA 1010-- FD selected not in COBOL program.

Condition Identifier: FA\$FD_NOT_IN_COBOL_PROGRAM Code: FA 1010

Description: The file description specified in the FD_NAME parameter of the COLLECT_FILE_DESCRIPTION command is not in the COBOL 5 program.

User Action: Verify that the correct file description was specified.

--ERROR FA 1020-- Assigned name selected not in COBOL program.

Condition Identifier: FA\$ASSIGN_NOT_IN_COBOL_PROGRAM Code: FA 1020

Description: The name specified in the ASSIGNED_NAME parameter of the COLLECT_FILE_DESCRIPTION command does not appear in a SELECT clause of the FILE-CONTROL paragraph for the file.

User Action: Verify that the proper name was specified in the ASSIGNED_NAME parameter.

--ERROR FA 1030-- Assigned name appears in more than 1 FD.

Condition Identifier: FA\$ASSIGN_IN_MORE_THAN_1_FD Code: FA 1030

Description: The name you specified in the ASSIGNED_NAME parameter of the COLLECT_FILE_DESCRIPTION appears in more than one file description.

User Action: Correct file description.

--ERROR FA 1040-- Assigned name selected does not match assigned name for the FD selected.

Condition Identifier: FAE\$ASSIGN_AND_FD_DO_NOT_MATCH Code: FA 1040

Description: The name specified in the ASSIGNED_NAME parameter of the COLLECT_FILE_DESCRIPTION command does not match the file description.

User Action: Correct the ASSIGNED_NAME parameter.

--ERROR FA 1050-- Neither an assigned_name nor an FD parameter was specified.

Condition Identifier: FAE\$ASSIGN_NOR_FD_SPECIFIED Code: FA 1050

Description: You must specify either the ASSIGNED_NAME or the FD_NAME parameter in the COLLECT_FILE_DESCRIPTION command.

User Action: Specify one of the parameters.

--ERROR FA 1060-- Selected record not a record in WORKING-STORAGE: {text}.

Condition Identifier: FAE\$SELECTED_REC_NOT_IN_PROGRAM Code: FA 1060

Description: The selected record is not in the COBOL program.

User Action: Verify that the proper record was specified.

--ERROR FA 1070-- Errors were detected by COLLECT_FILE_DESCRIPTION. See ERROR file for details.

Condition Identifier: FAE\$ERRORS_IN_COLFD Code: FA 1070

Description: The COLLECT_FILE_DESCRIPTION command detected errors in the COBOL5 input file description.

User Action: See the COLLECT_FILE_DESCRIPTION error file for specific errors. The default error file is \$ERRORS.

--ERROR FA 1080-- FMA cannot migrate files with this organization.

Condition Identifier: FAE\$CANNOT_MIGRATE_FO Code: FA 1080

Description: Because of its file organization, the file cannot be migrated using FMA.

User Action: Consider other migration methods. The data could be written to another 170 file in a file organization that is supported by FMA.

--ERROR FA 1090-- File named in parameter not found - {text}.

Condition Identifier: FAE\$IFD_FILE_NOT_FOUND Code: FA 1090

Description: The file you specified does not exist.

User Action: Verify that the proper file name was specified.

--ERROR FA 1100-- Fatal errors were found in the COBOL5 source program. See ERROR file for details.

Condition Identifier: FAE\$ERRORS_FROM_COBOL5 Code: FA 1100

Description: COBOL FMA requires a correct COBOL5 source program.

User Action: Correct errors in the COBOL5 program.

--ERROR FA 1110-- Errors were received for the NOSVE COBOL compiler.

Condition Identifier: FAE\$ERRORS_FROM_NOSVE_COBOL Code: FA 1110

Description: The NOS/VE COBOL compiler detected errors.

User Action: Correct the errors.

**--ERROR FA 1120-- Group to be converted contains subordinate REDEFINES:
{text}**

Condition Identifier: FAE\$REDEFINES_IN_GROUP Code: FA 1120

Description: The COBOL5 input file description contains a subordinate REDEFINES clause.

User Action: In order to migrate a file with a REDEFINES clause in its file description, you must write a record procedure that informs FMA which data field definition to use to migrate each record in the file. Record procedures are discussed in the migration usage manuals.

--ERROR FA 1130-- Errors detected in record procedure. See LIST file for details.

Condition Identifier: FAE\$RECORD_PROCEDURE_ERRORS Code: FA 1130

Description: The record procedure you wrote contains one or more errors. The specific errors are listed in the file specified by the LIST parameter on the MIGRATE_FILE command.

User Action: Examine the LIST file. If you did not specify a LIST file, you may need to run the migration task.

--ERROR FA 1140-- FMA does not allow SETFA of Key_Type = Integer.

Condition Identifier: FAE\$SETFA_KT_TO_INTEGER Code: FA 1140

Description: To migrate this file, you must specify KEY_TYPE=COLLATED or KEY_TYPE=UNCOLLATED on the SETFA command.

User Action: Change the KEY_TYPE parameter on the SETFA command to specify COLLATED or UNCOLLATED.

--ERROR FA 1150-- The required record procedure is absent, or insufficient due to unreferenced alternatives for multiple 01's or REDEFINES that exist in the input file description.

Condition Identifier: FAE\$RECORD_PROCEDURE_DEFICIENT Code: FA 1150

Description: The file you are migrating requires a record procedure to handle multiple 01 descriptors or REDEFINES clauses. If you provide a record procedure, it does not resolve all of the ambiguities created by the multiple 01s or REDEFINES clauses in the input field description.

User Action: If the record procedure is missing, supply one. If you have supplied one, make the necessary changes to resolve all multiple 01's or REDEFINES clauses in the file description.

--WARNING FA 1160-- One or more conversion references in the record procedure are redundant because they reference unambiguous parts of the input file description.

Condition Identifier: FAE\$RECORD_PROCEDURE_REDUNDANT Code: FA 1160

Description: Your record procedure references datanames that do not appear in a multiple 01 description or REDEFINES clause. These conversion references are not needed and are disregarded by FMA.

User Action: No action is necessary because FMA disregards the redundant conversion references. To eliminate the error, check the record procedure and eliminate the redundant references.

--WARNING FA 1170-- Result of COMP-1 item, {text}, can be read by COBOL program compiled with BL=COBOL5.

Condition Identifier: FAE\$COMP1_IN_170_FILE Code: FA 1170

Description: The COBOL5 file contains one or more COMP-1 items. To read the migrated file on NOS/VE, you should compile the COBOL program with BL=COBOL5 specified on the COBOL command. To read the file with the BL parameter omitted, you must make certain changes to the COBOL program.

User Action: Before reading the migrated file, either compile the NOS/VE COBOL program with BL=COBOL5, or make the following changes to the source program before compiling with the BL parameter omitted.

- Add a SYNCHRONIZED clause to the data item description.
- Ensure that the data item has enough digits to fill a NOS/VE 64-bit word.

--ERROR FA 1180-- Cannot find name: {text} .

Condition Identifier: FAE\$NAME_NOT_FOUND Code: FA 1180

Description: The specified name cannot be found.

User Action: Verify that the correct name was specified.

--WARNING FA 1190-- Record number {text} has {text} fields where content is incompatible with data type. Default values suitable to type have been substituted. The first {text} errors detected are in fields (first byte position of field/length of field in bytes): {text}

Condition Identifier: FAE\$CONVERSION_WARNING Code: FA 1190

Description: The indicated record contains values whose data type does not match the data type of the fields as defined in the source program. FMA has assigned default values to the corresponding fields in the output record.

User Action: Check the file description in the COBOL program. You may have specified the wrong file description, or the file may contain bad data.

--ERROR FA 1200-- A field referenced by a record procedure IF statement, or a DEPENDING field of an OCCURS, begins in byte {text}. This fails on record number {text} because the record has only {text} bytes of data.

Condition Identifier: FAE\$FIELD_ERROR Code: FA 1200

Description: The referenced field is outside of the current record.

User Action: Change the IF statement or DEPENDING field to reference a field within the record. If the record is a variable length record, you can check the record length by using the \$IRL function.

--ERROR FA 1210-- A field referenced by a record procedure IF statement, or a DEPENDING field of an OCCURS, starting in byte position {text}, has content incompatible with data type. Consult last warning in ERROR file for detail.

Condition Identifier: FAE\$RELATION_ERROR Code: FA 1210

Description: The content of a data field must match the data type of the field or be declared in the record description in the COBOL 5 program.

User Action: Examine the file specified by the ERROR parameter on the MIGRATE_FILE command. This file should contain a message identifying the field that caused the error.

--ERROR FA 1220-- The following WARNING diagnostic was issued when processing the 170 file: {text}.

Condition Identifier: FAE\$WARNING_FROM_170_FILE Code: FA 1220

Description: A warning-level error occurred while FMA was migrating the CYBER 170 file.

User Action: Examine the message text and correct the error.

--ERROR FA 1230-- The length of the field that contains the record length is {text} bytes. This is not compatible with the data type for the field.

Condition Identifier: FAE\$D_RECORD_FIELD_LENGTH_TYPE Code: FA 1230

Description: A conversion error occurred when FMA attempted to store the record length value into the field named in the DEPENDING ON phrase of a RECORD CONTAINS clause.

User Action: Check the data type of the record length field, it must be DISPLAY or COMP-1, and must not exceed six digits.

--ERROR FA 1240-- The record length field is not large enough to contain the record length value for record number {text}. The field length is {text} bytes. The record length is {text} bytes.

Condition Identifier: FAE\$D_RECORD_FIELD_OVERFLOW Code: FA 1240

Description: The record length of the indicated record exceeded the size of the record length field (the field named in the DEPENDING ON phrase of a RECORD CONTAINS clause).

User Action: Increase the size of the record length field (as defined in the COBOL5 program).

--WARNING FA 1250-- Migrate_File is reading a variable record file, but creating a fixed record file.

Condition Identifier: FAES\$VARIABLE_INPUT_FIXED_OUTPUT Code: FA 1250

Description: The MIGRATE_FILE command of COBOL FMA is migrating a file with variable length records to a file with fixed length records.

User Action: Verify that this is the desired situation. If so, ensure that the output record length is at least as big as the maximum record length of the file being migrated.

--FATAL FA 1260-- Sum of lengths for primary and alternate key exceeds 230 characters

Condition Identifier: FAES\$KEY_TOO_BIG Code: FA 1260

Description: You are using COBOL FMA to migrate a NOS/VE indexed sequential or direct access file to NOS. The NOS/VE file description has the lengths of the primary key and of the longest alternate key adding up to a sum exceeding 230. NOS cannot handle such long keys.

User Action: Change the description of the NOS/VE file in the NOS/VE COBOL program so that the lengths of the primary key and of the longest alternate key add up to a sum less than or equal to 230.

--FATAL FA 1261-- Requested output file is already permanent - {text}.

Condition Identifier: FAES\$PERM_FILE Code: FA 1261

Description: COBOL FMA is trying to migrate a NOS/VE file to a NOS permanent file. The NOS permanent file already exists and cannot be overwritten by the migrating NOS/VE file.

User Action: Change COBOL FMA to refer to a NOS permanent file that does not already exist.

--FATAL FA 9902-- File specified is not in catalog.

Condition Identifier: FAES\$FILE_NOT_IN_CATALOG Code: FA 9902

Description: You specified a file that could not be found in the specified catalog.

User Action: Check the file path specifications for an incorrect catalog name or file name.

--FATAL FA 9903-- File specified is not a tape request.

Condition Identifier: FAES\$FILE_NOT_TAPE_REQUEST Code: FA 9903

Description: For migrating a VAX tape file, the file specified in a SET_INPUT_ATTRIBUTES directive must be a tape request file. A tape request file is created by a CREATE_VAX_REQUEST command.

User Action: Change the file specified by the SET_INPUT_ATTRIBUTES directive to a file created by a CREATE_VAX_REQUEST command.

--FATAL FA 9904-- Duplicate external volume serial numbers (VSNs).

Condition Identifier: FAES\$DUPLICATE_EXTERNAL_VSNS Code: FA 9904

Description: The VOLUME_SERIAL_NUMBER (VSN) parameter on a CREATE_VAX_REQUEST (CREVR) command specified incorrect external volume serial numbers.

User Action: Change the CREATE_VAX_REQUEST command to specify the correct volume serial numbers.

--FATAL FA 9905-- Non ansi label headers or internal code incorrect.

Condition Identifier: FAES\$NON_ANSI_LABELS Code: FA 9905

Description: The tape label is nonstandard.

User Action: Use a tape with standard ANSI label headers.

--FATAL FA 9906-- File specified not in volume set.

Condition Identifier: FAES\$FILE_NOT_IN_VOLUME_SET Code: FA 9906

Description: The file specified does not exist.

User Action: Verify that the correct file name was specified.

--FATAL FA 9907-- Wrong tape was mounted or vsn list out of order.

Condition Identifier: FAE\$SECTION_OUT_OF_SEQUENCE Code: FA 9907

Description: The wrong tape was mounted by the operator or the volume serial number list is out of order.

User Action: Verify that the operator mounted the correct tape.

--FATAL FA 9908-- File set identifier conflicts with the volume.

Condition Identifier: FAE\$FILE_SET_ID_CONFLICT Code: FA 9908

Description: An improper file set identifier was specified.

User Action: Verify that the proper file set identifier was specified.

--FATAL FA 9909-- Unsupported record format.

Condition Identifier: FAE\$UNSUPPORTED_RECORD_FORMAT Code: FA 9909

Description: The specified record format is not supported.

User Action: No action. The file cannot be migrated.

--FATAL FA 9910-- Skip backward is unsupported for tape files.

Condition Identifier: FAE\$SKIP_BACKWARD_UNSUPPORTED Code: FA 9910

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--FATAL FA 9911-- Skip partition is unsupported for tape files.

Condition Identifier: FAE\$SKIP_PARTITION_UNSUPPORTED Code: FA 9911

Description: An internal error has occurred.

User Action: Follow site-defined procedure for reporting system problems.

--FATAL FA 9912-- Tape block size too large.

Condition Identifier: FAE\$BLOCK_SIZE_TOO_LARGE Code: FA 9912

Description: The block size specified for the tape exceeds the allowable maximum.

User Action: Specify a smaller block size.

--FATAL FA 9913-- Tape block size too small.

Condition Identifier: FAE\$BLOCK_SIZE_TOO_SMALL Code: FA 9913

Description: The block size specified for the tape is less than the minimum.

User Action: Specify a larger block size.

--FATAL FA 9914-- Command 'Create_vax_request' not issued.

Condition Identifier: FAE\$VAX_REQUEST_NOT_ISSUED Code: FA 9914

Description: You must specify a CREATE_VAX_REQUEST command before migrating a VAX tape file.

User Action: Include a CREATE_VAX_REQUEST command.

--FATAL FA 9915-- Specified internal vsn does not match with external vsn.

Condition Identifier: FAE\$INVALID_INTERNAL_VSN Code: FA 9915

Description: The internal and external volume serial numbers do not match.

User Action: Correct the internal volume serial number.

--FATAL FA 9916-- File sequence number {text} and file identifier {text} do not match for the specified file.

Condition Identifier: FAE\$FSN_AND_FI_NOT_MATCHING Code: FA 9916

Description: The file set does not contain a file having the specified file sequence number and file identifier.

User Action: Correct the file sequence number or file identifier.

--FATAL FA 9917-- Specified volume set not in file sequence order.

Condition Identifier: FAES\$VOLUME_OUT_OF_SEQUENCE Code: FA 9917

Description: Multiple tape volumes must be requested in order of file sequence number.

User Action: Correct the CREATE_VAX_REQUEST command so that the volumes are specified in the correct order.

--FATAL FA 9921-- Actual record length is larger than maximum record length (MAXRL, MRL).

Condition Identifier: FAES\$RECORD_LARGER_THAN_MRL Code: FA 9921

Description: Improperly specified value for MAXIMUM_RECORD_LENGTH parameter on a tape request command. Tape file has a record with length greater than the value of the MAXIMUM_RECORD_LENGTH parameter.

User Action: Increase the value specified by the MAXIMUM_RECORD_LENGTH parameter.

--FATAL FA 9922-- Max block length not large enough.

Condition Identifier: FAES\$MAX_BLOCK_LENGTH_SMALL Code: FA 9922

Description: Improperly specified value for MAXIMUM_BLOCK_LENGTH parameter on a tape migration command. Tape file has block with a length greater than the value of the MAXIMUM_BLOCK_LENGTH parameter.

User Action: Increase the value of the MAXIMUM_BLOCK_LENGTH parameter.

--FATAL FA 9924-- The tape file is either not W record or max record length is small.

Condition Identifier: FAES\$FILE_NOT_W_RECORD Code: FA 9924

Description: The value was improperly specified for either the RECORD_TYPE or the MAXIMUM_RECORD_LENGTH parameter of a tape migration command. EITHER: RECORD_TYPE=CONTROL_WORD and the tape file has a record with a different record type. OR: The tape file has a record with a length greater than the value of the MAXIMUM_RECORD_LENGTH parameter.

User Action: EITHER: Change the value of the RECORD_TYPE parameter from CONTROL_WORD. OR: Increase the value of the MAXIMUM_RECORD_LENGTH parameter.

--ERROR FA 9950-- File has already been opened, please use {text} command to change attributes.

Condition Identifier: FAES\$TAPE_FILE_ALREADY_OPEN Code: FA 9950

Description: A CREATE_xxx_REQUEST tape migration command tries to associate a temporary NOS/VE file with an xxx tape file. The temporary NOS/VE file is already associated with another tape file, that has been opened.

User Action: If the xxx tape file is on the same volume as the opened tape file, enter a CHANGE_xxx_REQUEST tape migration command to change the attributes of the xxx tape file associated with the temporary NOS/VE file. Otherwise, enter a CREATE_xxx_REQUEST command to associate the xxx tape file with a different temporary NOS/VE file.

--ERROR FA 9951-- Either External_VSN or Recorded_VSN parameter must be specified.

Condition Identifier: FAES\$TAPE_FILE_REQUIRES_VSN Code: FA 9951

Description: A CREATE_xxx_REQUEST tape migration command has specified neither the EXTERNAL_VSN nor the RECORDED_VSN parameter. NOS/VE has no way to identify the volume holding the tape file.

User Action: Specify a value for either the EXTERNAL_VSN or the RECORDED_VSN parameter.

--ERROR FA 9952-- File has not been opened yet, to change attributes you may use {text} command again.

Condition Identifier: FA\$TAPE_FILE_NOT_OPEN Code: FA 9952

Description: A CHANGE_XXX_REQUEST tape migration command tries to change the attributes of an xxx tape file associated with a temporary NOS/VE file. The temporary NOS/VE file has already been associated with an xxx tape file that has not been opened.

User Action: Because the xxx tape file has not been opened, you enter a CREATE_XXX_REQUEST tape migration command to change the attributes of the xxx tape file associated with the temporary NOS/VE file.

--ERROR FA 9953-- File is not on a magnetic tape drive, please use {text} command to set the initial attributes.

Condition Identifier: FA\$TAPE_FILE_NOT_MOUNTED Code: FA 9953

Description: A CHANGE_XXX_REQUEST tape migration command tries to change the attributes of an xxx tape file associated with a temporary NOS/VE file. The temporary NOS/VE file already exists and is associated with no tape files.

User Action: Because the temporary NOS/VE file is not already associated with a tape file, use the CREATE_XXX_REQUEST tape migration command to associate the xxx tape file with the temporary NOS/VE file.

--ERROR FA 9954-- The file is not the subject of a previously created {text} request.

Condition Identifier: FA\$TAPE_FILE_NOT_REQUESTED Code: FA 9954

Description: A CHANGE_YYY_REQUEST tape migration command tries to change the attributes of a yyy tape file associated with a temporary NOS/VE file. The temporary NOS/VE file is already associated with another file, an xxx tape file.

User Action: If you wish to change the attributes of a yyy tape file, you need to associate it with a different temporary NOS/VE file. If the yyy tape file has not been opened, use a CREATE_YYY_REQUEST tape migration command. If the yyy tape file has been opened, enter a CHANGE_YYY_REQUEST command that uses the temporary NOS/VE file associated with the opened yyy tape file. If you wish to change the attributes of the xxx tape file associated with the original temporary NOS/VE file, do the following: If the xxx tape file has not been opened, use a CREATE_XXX_REQUEST command. If the xxx tape file has been opened, use a CHANGE_XXX_REQUEST command.

--ERROR FA 9955-- Block and record type combination is not legal.

Condition Identifier: FA\$BT_AND_RT_ILLEGAL_FOR_TAPE Code: FA 9955

Description: A CREATE_XXX_REQUEST or CHANGE_XXX_REQUEST tape migration command has specified an illegal combination of block type and record type. The legal combinations are: Block type character-count with record types, fixed-length, system-record, or zero-byte; block type internal with record type control-word.

User Action: Enter the tape migration command with a legal combination of block type and record type.

--FATAL FA 9960-- The NOS/VE file must have record type attribute of UNDEFINED.

Condition Identifier: FA\$RECORD_MUST_BE_UNDEFINED Code: FA 9960

Description: A NOS/VE file that contains either an ETA file or a NOS or NOS/BE file must have the NOS/VE file attribute Record_Type=undefined. Note: This is not the same record type attribute that is specified via the SETEFA or SETIFA comments.

User Action: Change the NOS/VE file attribute.

--FATAL FC 121-- Input file {text} is empty or mispositioned -- compilation terminated.

Condition Identifier: FCE\$CFS_EMPTY_INPUT Code: FC 121

Description: The file specified by the input command parameter did not have a Fortran program.

User Action: Verify contents of the input file.

--FATAL FC 3020-- One or more files contain inconsistent or invalid attributes -- NO COMPILATION. See job log for further information.

Condition Identifier: FCE\$BAD_FILE_ATTRIBUTES Code: FC 3020

Description: You specified an invalid file as the source input file to the FORTRAN compiler. The source input file must be a text file containing FORTRAN source statements.

User Action: Check the file you specified for the INPUT parameter on the FORTRAN command.

--ERROR FD 1-- Terminal disconnected.

Condition Identifier: FDE\$TERMINAL_DISCONNECTED Code: FD 1

Description: The program cannot communicate with the terminal.

User Action: Wait until the terminal can communicate with the program or terminate the program.

--ERROR FD 5-- Form {text} does not exist.

Condition Identifier: FDE\$UNKNOWN_FORM_NAME Code: FD 5

Description: The specified form name cannot be found using the object code libraries in your command list.

User Action: Create the form or add the appropriate object code library to your command list

--ERROR FD 7-- Space is not available.

Condition Identifier: FDE\$NO_SPACE_AVAILABLE Code: FD 7

Description: Storage cannot be allocated for the required tables or record areas.

User Action: Terminate the program and then reduce the number of defined items for the form.

--ERROR FD 9-- Form identifier {text} does not exist.

Condition Identifier: FDE\$INVALID_FORM_IDENTIFIER Code: FD 9

Description: The form identifier is less than zero and greater than the fdc\$maximum_form_identifier (1000), or has not been assigned by a prior screen formatting request.

User Action: Change the value of the form identifier.

--ERROR FD 11-- Variable {text} does not exist for form {text}.

Condition Identifier: FDE\$UNKNOWN_VARIABLE_NAME Code: FD 11

Description: The variable name does not exist in the form definition.

User Action: Change the variable name to match one in the form definition.

--ERROR FD 12-- Too many integer ranges specified for validation for form {text}.

Condition Identifier: FDE\$TOO_MANY_INTEGERS Code: FD 12

Description: A variable has too many integer ranges defined for terminal user validation.

User Action: Reduce the number of integer ranges defined for validation. These must not be greater than fdc\$maximum_valid_ranges (10,000).

--ERROR FD 13-- Object {text} already exists for form {text}.

Condition Identifier: FDE\$OBJECT_NAME_EXISTS Code: FD 13

Description: You have already used the specified object name in the form definition.

User Action: Specify a unique object name.

--ERROR FD 14-- Program work area does not match {text} form definition.

Condition Identifier: FDE\$WORK_AREA_INVALID Code: FD 14

Description: The length of the record (work area) in the program is not equal to the length of the record of the form definition on the object code library.

User Action: Generate a new record definition from the form definition on the object code library. Copy the record definition to your program.

--ERROR FD 17-- Width {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_WIDTH Code: FD 17

Description: The width is less than 1, greater than fdc\$maximum_x_position (256), or is outside the defined area of the form.

User Action: Change the width to be greater than or equal to 1 and less than or equal to fdc\$maximum_x_position.

--ERROR FD 18-- Height {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_HEIGHT Code: FD 18

Description: The height is less than 1, greater than fdc\$maximum_y_position (256), or is outside the defined area of the form.

User Action: Change the height to be greater than or equal to 1 or less than or equal to fdc\$maximum_y_position.

--ERROR FD 19-- Message form {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_MESSAGE_FORM_NAME Code: FD 19

Description: The message form name does not comply with the rules for a name under the System Command Language.

User Action: Change the message form name to comply with the System Command Language rules.

--ERROR FD 20-- Occurrence {text} is not valid for object {text} form {text}.

Condition Identifier: FDE\$INVALID_OCCURRENCE Code: FD 20

Description: The occurrence is less than 1 or greater than fdc\$maximum_occurrence (1000).

User Action: Change the occurrence to be greater than or equal to 1 and less than or equal to fdc\$maximum_occurrence.

--ERROR FD 21-- Character position {text} is not valid for object {text} form {text}.

Condition Identifier: FDE\$INVALID_CHARACTER_POSITION Code: FD 21

Description: The character position is less than 1 or greater than the form definition length for the object.

User Action: Examine the form definition to learn the length of the object. Change the character position to be greater than or equal to 1 and less than or equal to the length of the object.

--ERROR FD 22-- Invalid mode for variable {text}, form {text}.

Condition Identifier: FDE\$INVALID_MODE Code: FD 22

Description: The input/output mode for a variable is not one given by the type fdt\$io_mode.

User Action: Use one of the input/output mode parameters specified by fdt\$io_mode.

--ERROR FD 25-- Object {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_OBJECT_NAME Code: FD 25

Description: The object name does not comply with the rules for a name under the form definition programming language (COBOL or CYBIL).

User Action: Change the object name to comply with the programming language rules.

--ERROR FD 26-- Form {text} is not a valid name.

Condition Identifier: FDE\$INVALID_FORM_NAME Code: FD 26

Description: The form name does not comply with the rules for a name under the form System Command Language.

User Action: Change the form name to comply with the System Command Language rules.

--ERROR FD 27-- Form {text} is not open.

Condition Identifier: FDE\$FORM_CLOSED Code: FD 27

Description: You attempted a program interface call on a form that is not currently open.

User Action: Open the form. Then call the desired program interface.

--ERROR FD 29-- Attribute {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_ATTRIBUTE_NAME Code: FD 29

Description: The display attribute name does not comply with the rules for a name under the form definition programming language (COBOL or CYBIL).

User Action: Change the display attribute to comply with the programming language rules.

--ERROR FD 33-- Object {text} does not exist for form {text}.

Condition Identifier: FDE\$UNKNOWN_OBJECT_NAME Code: FD 33

Description: The object name does not exist in the form definition.

User Action: Examine and use an object name that exists in the form definition.

--ERROR FD 35-- {text} is a not a valid name.

Condition Identifier: FDE\$INVALID_NAME Code: FD 35

Description: The name is less than 1 character or greater than osc\$max_name_size (31) characters.

User Action: Use a name that is 1 - 31 characters.

--ERROR FD 36-- System error {text} on form {text}.

Condition Identifier: FDE\$SYSTEM_ERROR Code: FD 36

Description: Some internal error occurred.

User Action: Write a PSR.

--ERROR FD 37-- Table {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_TABLE_NAME Code: FD 37

Description: The table name does not comply with the rules for a name under the form definition programming language (COBOL or CYBIL).

User Action: Change the table name to comply with the programming language rules.

--ERROR FD 38-- Variable {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_VARIABLE_NAME Code: FD 38

Description: The variable name does not comply with the rules for a name under the form definition programming language (COBOL or CYBIL).

User Action: Change the variable name to comply with the programming language rules.

--ERROR FD 39-- Pushed form {text} cannot be manipulated.

Condition Identifier: FDE\$FORM_PUSHED Code: FD 39

Description: The form is currently pushed and the specified request cannot be done on a pushed form.

User Action: Change the program so that the request is not issued on pushed form.

--ERROR FD 40-- Table {text} does not exist for form {text}.

Condition Identifier: FDE\$UNKNOWN_TABLE_NAME Code: FD 40

Description: The table name specified by the program does not exist in the form definition.

User Action: Examine and use a table name that has been defined for the form definition.

--ERROR FD 42-- There are no forms to pop.

Condition Identifier: FDE\$NO_FORMS_TO_POP Code: FD 42

Description: The program attempted to pop forms when no forms were added.

User Action: Change the program to delete the pop forms request.

--ERROR FD 45-- Form definition errors.

Condition Identifier: FDE\$FORM_DEFINITION_ERRORS Code: FD 45

Description: The form has errors and a form record definition cannot be generated.

User Action: Correct the form errors and then generate a record definition.

--ERROR FD 46-- No forms are scheduled to push.

Condition Identifier: FDE\$NO_FORMS_TO_PUSH Code: FD 46

Description: The program did a push forms request when no forms were added since the last push forms request.

User Action: Change the program and delete the push forms request.

--ERROR FD 52-- Form {text} has no variables.

Condition Identifier: FDE\$FORM_HAS_NO_VARIABLES Code: FD 52

Description: The program is attempting to get or replace some variables on the form, but the form has no variables defined.

User Action: Remove the request from the program or add variables to the form definition.

--ERROR FD 53-- No forms are scheduled to show.

Condition Identifier: FDE\$NO_FORMS_TO_SHOW Code: FD 53

Description: No forms have been scheduled for display.

User Action: Change the program to add forms before reading or showing forms.

--ERROR FD 54-- Form {text} instance is not scheduled.

Condition Identifier: FDE\$FORM_NOT_SCHEDULED Code: FD 54

Description: A request has been made on a form that is not currently added. The request requires that the form is added.

User Action: Change the program to add the form before executing the current request.

--ERROR FD 55-- Event {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_EVENT_NAME Code: FD 55

Description: The event name does not comply with the rules for a name under the form definition programming language.

User Action: Change the event name to comply with the programming language rules.

--ERROR FD 56-- X position {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_X_POSITION Code: FD 56

Description: The x position is less than 1, greater than fdc\$maximum_x_position (256), or is outside the defined area of the form.

User Action: Change the x position to be greater than or equal to 1 and less than or equal to fdc\$maximum_x_position.

--ERROR FD 57-- Y position {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_Y_POSITION Code: FD 57

Description: The y position is less than 1, greater than fdc\$maximum_y_position (256), or is outside the defined area of the form.

User Action: Change the y position to be greater than or equal to 1 or less than or equal to fdc\$maximum_y_position.

--ERROR FD 58-- Event {text} does not exist for form {text}.

Condition Identifier: FDE\$UNKNOWN_EVENT_NAME Code: FD 58

Description: The event name specified by the program does not exist in the form definition.

User Action: Examine and change the program to use one of the event names defined in the form definition.

--ERROR FD 59-- Deck {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_DECK_NAME Code: FD 59

Description: The deck name does not comply with the rules for a name under the form System Command Language.

User Action: Change the deck name to comply with the System Command Language rules.

--ERROR FD 60-- Record {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_RECORD_NAME Code: FD 60

Description: The record name does not comply with the rules for a name under the form definition programming language.

User Action: Change the record name to comply with the programming language rules.

--ERROR FD 62-- Table {text} already exists for form {text}.

Condition Identifier: FDE\$TABLE_NAME_EXISTS Code: FD 62

Description: The table name you have chosen has already been used.

User Action: Choose a unique table name.

--ERROR FD 63-- Object at x position {text}, y position {text}, overlays an existing object on form {text}.

Condition Identifier: FDE\$OBJECT_OVERLAYS Code: FD 63

Description: The object you are creating would overlay an existing object at the specified x,y position.

User Action: Choose a form location that has no objects.

--ERROR FD 64-- Too many ranges specified for validation of real numbers for form {text}.

Condition Identifier: FDE\$TOO_MANY_REALS Code: FD 64

Description: A variable has too many real ranges defined for terminal user validation.

User Action: Reduce the number of real ranges defined for validation. These must not be greater than fdc\$maximum_valid_ranges (10,000).

--ERROR FD 65-- Too many strings specified for validation of strings for form {text}.

Condition Identifier: FDE\$TOO_MANY_STRINGS Code: FD 65

Description: A variable has too many valid strings defined for terminal user validation.

User Action: Reduce the number of strings used for validation. These must not be greater than fdc\$maximum_valid_strings (10,000).

--ERROR FD 66-- No object at x position {text}, y position {text} for form {text}.

Condition Identifier: FDE\$NO_OBJECT_AT_POSITION Code: FD 66

Description: No object exists at the specified x,y position.

User Action: Examine the form definition for objects and select the correct x,y position for the desired object.

--ERROR FD 68-- String is too small.

Condition Identifier: FDE\$STRING_TOO_SMALL Code: FD 68

Description: The storage area for a string parameter is not large enough to hold the requested form definition attribute.

User Action: Examine the form definition and allocate a larger area for the requested form definition attribute.

--ERROR FD 69-- Variable {text} already exists for form {text}.

Condition Identifier: FDE\$VARIABLE_NAME_EXISTS Code: FD 69

Description: The variable name you have chosen has already been used.

User Action: Choose a unique variable name.

--ERROR FD 70-- Form {text} is already added.

Condition Identifier: FDE\$FORM_ALREADY_ADDED Code: FD 70

Description: The form was previously added by the program.

User Action: Change the program so duplicate adds of a form are not repeated.

--ERROR FD 73-- Open form, form {text}, cannot be updated.

Condition Identifier: FDE\$CANNOT_UPDATE_OPENED_FORM Code: FD 73

Description: Requesting a form definition change cannot be done on an open form.

User Action: Copy the open form, issue an edit form request on the copied form, and then issue the change request.

--ERROR FD 78-- Display {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_DISPLAY_NAME Code: FD 78

Description: The display attribute name does not comply with rules for a name under the form definition programming language (COBOL or CYBIL).

User Action: Change the display attribute name to comply with the programming language rules.

--ERROR FD 79-- Real range {text} - {text} is not valid for variable {text}, form {text}.

Condition Identifier: FDE\$INVALID_REAL_RANGE Code: FD 79

Description: The minimum real number must be less than or equal to the maximum real number.

User Action: Change the ranges given for the valid real numbers.

--ERROR FD 80-- Integer range {text} - {text} is not valid variable {text}, form {text}.

Condition Identifier: FDE\$INVALID_INTEGER_RANGE Code: FD 80

Description: The minimum integer number must be less than or equal to the maximum integer number.

User Action: Change the ranges given for the valid integer numbers.

--ERROR FD 81-- Integer range {text} - {text} is not known for variable {text}, form {text}.

Condition Identifier: FDE\$UNKNOWN_INTEGER_RANGE Code: FD 81

Description: The program attempted to delete an integer range validation that does not exist.

User Action: Examine and choose an integer range that exists in the form definition.

--ERROR FD 82-- Real range {text} - {text} does not exist for variable {text}, form {text}.

Condition Identifier: FDE\$UNKNOWN_REAL_RANGE Code: FD 82

Description: The program attempted to delete a real range validation that does not exist.

User Action: Examine and choose a real range that exists in the form definition.

--ERROR FD 83-- Validation string {text} does not exist for variable {text}, form {text}.

Condition Identifier: FDE\$UNKNOWN_VALID_STRING Code: FD 83

Description: The program attempted to delete a string validation that does not exist.

User Action: Examine and choose a string that exists in the form definition.

--ERROR FD 84-- Display {text} already exists for form {text}.

Condition Identifier: FDE\$DISPLAY_NAME_EXISTS Code: FD 84

Description: The chosen display attribute name already exists.

User Action: Choose a unique display attribute name.

--ERROR FD 85-- Event {text} already exists for form {text}.

Condition Identifier: FDE\$EVENT_NAME_EXISTS Code: FD 85

Description: The chosen event name already exists in the form definition.

User Action: Choose a unique event name.

--ERROR FD 86-- Display {text} does not exist for form {text}.

Condition Identifier: FDE\$UNKNOWN_DISPLAY_NAME Code: FD 86

Description: The display attribute name does not exist in the form definition.

User Action: Examine and choose the desired display attribute name defined in the form definition.

--ERROR FD 87-- There are too many form names.

Condition Identifier: FDE\$TOO_MANY_FORM_NAMES Code: FD 87

Description: The program did not allocate enough storage to hold all the names for the form definition.

User Action: Allocate a large space for form names. The fdp\$get_form_attributes request tells you the number of form names.

--ERROR FD 88-- There are too many object names for form {text}.

Condition Identifier: FDE\$TOO_MANY_FORM_OBJECTS Code: FD 88

Description: The program did not allocate enough storage to hold all the objects in the form definition.

User Action: Allocate a larger space for form objects. The fdp\$get_form_attributes request tells you the number of form objects.

--ERROR FD 89-- No text at x equals {text}, y equals {text}, form {text}.

Condition Identifier: FDE\$NO_TEXT_AT_POSITION Code: FD 89

Description: No text object exists at the specified x,y positions.

User Action: Examine and change the x,y positions for the desired text as defined in form definition.

--ERROR FD 90-- Object {text} requires text for form {text}.

Condition Identifier: FDE\$NO_TEXT_FOR_OBJECT Code: FD 90

Description: The form object requires a text definition.

User Action: Specify text for the object.

--ERROR FD 91-- Occurrence {text} does not exist for object {text} form {text}.

Condition Identifier: FDE\$UNKNOWN_OCCURRENCE Code: FD 91

Description: The occurrence does not exist in the form definition.

User Action: Examine and specify the correct occurrence for the variable as defined in the form definition.

--ERROR FD 93-- There is a range {text} - {text} overlap for variable {text}, form {text}.

Condition Identifier: FDE\$RANGE_OVERLAP Code: FD 93

Description: The specified range for integers or reals overlap and existing range in the form definition.

User Action: Examine and choose a range that exists in the form definition.

--ERROR FD 94-- There are no comments to delete for form {text}.

Condition Identifier: FDE\$NO_COMMENTS_TO_DELETE Code: FD 94

Description: The program is attempting to delete comments from a form definition. No comments exist.

User Action: Remove the request to delete comments.

--ERROR FD 95-- Occurrence {text} already exists for object {text} for form {text}.

Condition Identifier: FDE\$OBJECT_OCCURRENCE_EXISTS Code: FD 95
Description: The chosen object occurrence already exists in the form definition.
User Action: Choose a unique object occurrence.

--ERROR FD 97-- Validation string {text} already exists for variable {text}, form {text}.

Condition Identifier: FDE\$VALID_STRING_EXISTS Code: FD 97
Description: The string of characters specified for validation already exists in the form definition.
User Action: Remove the request to add a new character string validation.

--ERROR FD 98-- Object change {text} is not valid.

Condition Identifier: FDE\$INVALID_OBJECT_CHANGE Code: FD 98
Description: The program is specifying an attribute change that does not apply to the object.
User Action: Examine and choose an attribute that applies to the object as defined in the form definition.

--ERROR FD 99-- Address is not valid in request for form {text}.

Condition Identifier: FDE\$INVALID_ADDRESS Code: FD 99
Description: The application program specified a NIL value for pointer.
User Action: Use a non-NIL pointer.

--ERROR FD 100-- Terminal name does not exist in terminal definitions. Use SET_TERMINAL_ATTRIBUTES command with parameter TERMINAL_NAME.

Condition Identifier: FDE\$TERMINAL_NOT_IDENTIFIED Code: FD 100
Description: Your terminal name does not exist in the terminal definitions.
User Action: Change terminal attributes specifying TERMINAL_MODE equal to a terminal name that is defined.

--ERROR FD 101-- Form language {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_FORM_LANGUAGE Code: FD 101
Description: The form programming language is not one specified by fdt\$form_processor.
User Action: Use one of the programming languages specified by fdt\$form_processor.

--ERROR FD 102-- Form area key is not valid for form {text}.

Condition Identifier: FDE\$INVALID_FORM_AREA_KEY Code: FD 102
Description: The form area key must be one of those specified by fdt\$form_area_key.
User Action: Use one of the keys given by fdt\$form_area_key.

--ERROR FD 103-- A form name is required to save a form.

Condition Identifier: FDE\$FORM_NAME_REQUIRED Code: FD 103
Description: To save a form, the form name must have been previously specified.
User Action: Give the form a name using the change form attributes request.

--ERROR FD 104-- No forms scheduled to read.

Condition Identifier: FDE\$NO_FORMS_TO_READ Code: FD 104
Description: To do the read forms request at least one form must be added.
User Action: Change the program to add a form before doing the read forms request.

--ERROR FD 105-- Help form {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_HELP_FORM_NAME Code: FD 105

Description: The name of the help form does not comply with the rules for a name under the System Command Language.

User Action: Change the program to specify a name that complies with the System Command Language rules.

--ERROR FD 106-- Error form {text} is not a valid name for form {text}.

Condition Identifier: FDE\$INVALID_ERROR_FORM_NAME Code: FD 106

Description: The name of the error form does not comply with the rules for a name under the System Command Language.

User Action: Change the program to specify a name that complies with the System Command Language rules.

--ERROR FD 107-- Create mark is only valid on a design form. Form is {text}.

Condition Identifier: FDE\$CREATE_MARK_INVALID Code: FD 107

Description: A create mark request is only valid on a design form.

User Action: Delete the create mark request from the program or issue the request on a design form.

--ERROR FD 108-- Delete mark is only valid on a design form. Form is {text}.

Condition Identifier: FDE\$DELETE_MARK_INVALID Code: FD 108

Description: A delete mark request is only valid on a design form.

User Action: Remove the delete mark request from the program or issue the request on a design form.

--ERROR FD 109-- No mark defined for form {text}.

Condition Identifier: FDE\$NO_MARK_DEFINED Code: FD 109

Description: To delete a mark, a mark must currently exist. The mark does not currently.

User Action: Change the program so that a create mark request precedes a delete mark request.

--ERROR FD 110-- Area cuts object at x = {text}, y = {text} on form {text}.

Condition Identifier: FDE\$AREA_CUTS_OBJECT Code: FD 110

Description: The area specified by the request cannot split an object on the form definition.

User Action: Change the program so that no object is split.

--ERROR FD 111-- Copy area outside of form {text}.

Condition Identifier: FDE\$COPY_OUTSIDE_FORM Code: FD 111

Description: The program cannot copy an area that lies outside the area occupied by the form.

User Action: Examine the form definition to learn the area of the form, and change the request to copy an area inside the form.

--ERROR FD 112-- Move area outside of form {text}.

Condition Identifier: FDE\$MOVE_OUTSIDE_FORM Code: FD 112

Description: The program cannot move an area that lies outside the area occupied by the form.

User Action: Examine the form definition to learn the area occupied by the form, and change the request to move an area inside the form.

--ERROR FD 113-- Invalid form attribute for form {text}.

Condition Identifier: FDE\$INVALID_FORM_ATTRIBUTE Code: FD 113

Description: The program specified a form attribute that is not given by the type fdt\$change_form_key or fdt\$get_form_key.

User Action: Change the program to use one of the keys specified by fdt\$change_form_key or fdt\$get_form_key.

--ERROR FD 114-- Invalid record attribute for form {text}.

Condition Identifier: FDE\$INVALID_RECORD_ATTRIBUTE Code: FD 114

Description: The program specified a record attribute that is not given by the type fdt\$change_record_key or fdt\$get_record_key.

User Action: Change the program to use one of the keys specified by fdt\$change_form_key or fdt\$get_form_key.

--ERROR FD 115-- Invalid object key for form {text}.

Condition Identifier: FDE\$INVALID_OBJECT_KEY Code: FD 115

Description: The program specified an object attribute that is not given by the type fdt\$object_definition_key.

User Action: Change the program to use one of the keys specified by fdt\$object_definition_key.

--ERROR FD 116-- Invalid object attribute for form {text}.

Condition Identifier: FDE\$INVALID_OBJECT_ATTRIBUTE Code: FD 116

Description: The program specified an object attribute that is not given by the type fdt\$change_object_key or fdt\$get_object_key.

User Action: Change the program to use one of the keys specified by fdt\$change_object_key or fdt\$get_object_key.

--ERROR FD 117-- Invalid table attribute for form {text}.

Condition Identifier: FDE\$INVALID_TABLE_ATTRIBUTE Code: FD 117

Description: The program specified a table attribute that is not given by the type fdt\$change_table_key or fdt\$get_table_key.

User Action: Change the program to use one of the keys specified by fdt\$change_table_key or fdt\$get_table_key.

--ERROR FD 118-- Invalid program data type for form {text}.

Condition Identifier: FDE\$PROGRAM_DATA_TYPE Code: FD 118

Description: The program specified a program data type that is not given by the type fdt\$program_data_type.

User Action: Change the program to use one of the keys specified by fdt\$program_data_type.

--ERROR FD 119-- Invalid output format key for form {text}.

Condition Identifier: FDE\$INVALID_OUTPUT_FORMAT_KEY Code: FD 119

Description: The program specified an output format that is not given by the type fdt\$output_format_key.

User Action: Change the program to use one of the keys specified by fdt\$output_format_key.

--ERROR FD 120-- Invalid error key for form {text}.

Condition Identifier: FDE\$INVALID_ERROR_KEY Code: FD 120

Description: The program specified error processing that is not given by the type fdt\$error_key.

User Action: Change the program to use one of the keys specified by fdt\$error_key.

--ERROR FD 121-- Invalid variable attribute for form {text}.

Condition Identifier: FDE\$INVALID_VARIABLE_ATTRIBUTE Code: FD 121

Description: The program specified a variable attribute that is not given by the type fdt\$change_variable_key or fdt\$get_variable_key.

User Action: Change the program to use one of the keys specified by fdt\$change_variable_key or fdt\$get_variable_key.

--ERROR FD 123-- Invalid help key for form {text}.

Condition Identifier: FDE\$INVALID_HELP_KEY Code: FD 123

Description: The program specified help processing that is not given by the type fdt\$help_key.

User Action: Change the program to use one of the keys specified by fdt\$help_key.

--ERROR FD 124-- Feature {text} is not implemented.

Condition Identifier: FDE\$FEATURE_NOT_IMPLEMENTED Code: FD 124

Description: The program used a feature that is not completely implemented.

User Action: Change the program to remove the use of the unimplemented feature.

--ERROR FD 125-- Form {text}, cannot be changed.

Condition Identifier: FDE\$CANNOT_CHANGE_FORM Code: FD 125

Description: The form definition cannot be changed.

User Action: Change the program to use a create form or edit form request before attempting to change a form definition.

--ERROR FD 126-- Invalid record type for form {text}.

Condition Identifier: FDE\$INVALID_RECORD_TYPE Code: FD 126

Description: The program specified a record type that is not given by the type fdt\$record_type.

User Action: Change the program to use one of the keys specified by fdt\$record_type.

--ERROR FD 127-- Object at x position {text}, y position {text} is not inside form {text}.

Condition Identifier: FDE\$OBJECT_NOT_IN_FORM Code: FD 127

Description: The program attempted to define an object that does not lie inside the form.

User Action: Examine the form definition to learn the area occupied by the form. Change the x,y position of the object so that it lies inside the form.

--ERROR FD 128-- Invalid form processor for form {text}.

Condition Identifier: FDE\$INVALID_FORM_PROCESSOR Code: FD 128

Description: The program specified a form processing language that is not given by the type fdt\$form_processing.

User Action: Change the program to use one of the keys specified by fdt\$form_processor.

--ERROR FD 129-- X increment {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_X_INCREMENT Code: FD 129

Description: The x increment is less than zero or greater than fdc\$maximum_x_position (256).

User Action: Change the program to use an x increment that is greater than or equal to zero and less than or equal to fdc\$maximum_x_position.

--ERROR FD 130-- Y increment {text} is not valid for form {text}.

Condition Identifier: FDE\$INVALID_Y_INCREMENT Code: FD 130

Description: The y increment is less than zero or greater than fdc\$maximum_y_position (256).

User Action: Change the program to use a y increment that is greater than or equal to zero and less than or equal to fdc\$maximum_y_position.

--ERROR FD 131-- Form {text} at x={text}, y={text}, width={text}, height={text} outside of screen.

Condition Identifier: FDE\$FORM_TOO_LARGE_FOR_SCREEN Code: FD 131

Description: The defined area for a form is larger than the terminal screen.

User Action: Reduce the defined area of the form or use a terminal that has a larger screen.

--ERROR FD 132-- Invalid text processing for form {text}.

Condition Identifier: FDE\$INVALID_TEXT_PROCESSING Code: FD 132

Description: The program specified text processing that is not given by the type fdt\$text_box_processing.

User Action: Change the program to use one of the keys specified by fdt\$text_box_processing.

--ERROR FD 133-- Form {text} is not a design form.

Condition Identifier: FDE\$INVALID_DESIGN_FORM Code: FD 133

Description: The program attempted to use a request valid only on a form that is not a design form.

User Action: Change the program and remove the request or use a design form.

--ERROR FD 134-- No variable defined for object {text}, form {text}.

Condition Identifier: FDE\$NO_OBJECT_VARIABLE_DEFINED Code: FD 134

Description: The program attempted to set the cursor on an object that is not a variable.

User Action: Change the program to set the cursor on a variable object.

--ERROR FD 136-- Form {text} was not ended.

Condition Identifier: FDE\$FORM_NOT_ENDED Code: FD 136

Description: The program attempted to open a form whose definition was not completed by an end form request.

User Action: Change the program to do an end form request before opening the form.

--ERROR FD 137-- Form {text} has invalid event form name {text}.

Condition Identifier: FDE\$INVALID_EVENT_FORM_NAME Code: FD 137

Description: The error form name does not comply with rules for a name defined under the System Command Language.

User Action: Change the error form name to comply with the System Command Language rules.

--ERROR FD 138-- Form {text} has invalid event form definition key.

Condition Identifier: FDE\$INVALID_EVENT_FORM_KEY Code: FD 138

Description: The program attempted to use event processing that is not given by the type fdt\$event_form_key.

User Action: Change the program to use one of the keys specified by fdt\$event_form_key.

--ERROR FD 139-- Form {text} is already open.

Condition Identifier: FDE\$FORM_ALREADY_OPEN Code: FD 139

Description: The program attempted to open a form that is already open.

User Action: Change the program to remove the duplicated open request.

--ERROR FD 141-- Form {text} requires conversion.

Condition Identifier: FDE\$FORM_REQUIRES_CONVERSION Code: FD 141

Description: The object code library data structures for a form definition have changed.

User Action: The form needs conversion to the new object code library data structures.

--ERROR FD 142-- Screen must have some active events.

Condition Identifier: FDE\$NO_EVENTS_ACTIVE Code: FD 142

Description: To do a read forms request, at least one event must be active on the screen.

User Action: Use a show forms request, define an event for a form, or add a form that has an event defined.

--ERROR FD 143-- Delete area outside of form {text}.

Condition Identifier: FDE\$DELETE_OUTSIDE_FORM Code: FD 143

Description: The program attempted to delete an area that does not lie within the form.

User Action: Examine the form definition and specify an area that lies with the form.

--ERROR FD 144-- Mark area outside of form {text}.

Condition Identifier: FDE\$MARK_OUTSIDE_FORM Code: FD 144

Description: The program attempted to mark an area that does not lie within the form.

User Action: Examine the form definition and specify an area that lies with the form.

--ERROR FD 145-- Bad data value.

Condition Identifier: FDE\$BAD__DATA_VALUE Code: FD 145

Description: The program used an invalid address or value for a parameter.

User Action: Change the invalid parameter in the request.

--ERROR FD 146-- Cannot produce a FORTRAN record definition for form {text}.

Condition Identifier: FDE\$RECORD_DEFN_NOT_WRITTEN Code: FD 146

Description: A FORTRAN record definition cannot be produced for a form with a non-FORTRAN process and one of the tables has more than one variable.

User Action: Create a new form with FORTRAN as the form processor.

--ERROR FD 147-- This command expects a variable type of {text}.

Condition Identifier: FDE\$WRONG_VARIABLE_TYPE Code: FD 147

Description: One of the following data types does not agree with the data type defined for the form:
FDP\$GET_REAL_VARIABLE, FDB\$GET_STRING_VARIABLE, FDBETR, FDGETI, FDGETS,
FDP\$XGET_REAL_VARIABLE, FDP\$XGET_INTEGER_VARIABLE, FDP\$XGET_STRING_VARIABLE.

User Action: Use a request that matches the data type.

--ERROR FD 148-- Variable {text} has invalid length {text} for form {text}.

Condition Identifier: FDE\$INVALID_VARIABLE_LENGTH Code: FD 148

Description: The length specified for a variable exceeds the maximum specified.

User Action: Respecify the length of the variable within the expected range.

--ERROR FD 149-- Event trigger already exists for form {text}.

Condition Identifier: FDE\$EVENT_TRIGGER_EXISTS Code: FD 149

Description: The specified event trigger has already been used.

User Action: Do not duplicate an event trigger definition.

--ERROR FD 150-- Form {text} is already combined.

Condition Identifier: FDE\$FORM_ALREADY_COMBINED Code: FD 150

Description: You attempted to combine a form which is already combined.

User Action: Remove the duplicate call to combine a form.

--ERROR FD 151-- Invalid table size for table {text}, form {text}.

Condition Identifier: FDE\$INVALID_TABLE_SIZE Code: FD 151

Description: The number of stored occurrences must be greater than or equal to the number of visible occurrences.

User Action: Change the number of stored occurrences to be greater than or equal to the number of visible occurrences.

--ERROR FD 152-- Form {text} not added.

Condition Identifier: FDE\$FORM_NOT_ADDED Code: FD 152

Description: The added form specified by the combine form request has not been previously added.

User Action: Add the form. Then use the combine form request.

--ERROR FD 153-- Invalid input format key for form {text}.

Condition Identifier: FDE\$INVALID_INPUT_FORMAT_KEY Code: FD 153

Description: The input format key for a variable must match one of those given by fdt\$input_format.

User Action: None.

--INFORMATIVE FD 154-- Please correct.

Condition Identifier: FDE\$SYSTEM_ERROR_MESSAGE Code: FD 154

Description: The terminal user entered an invalid data for a variable. The form definition used the default error message.

User Action: See help.

--INFORMATIVE FD 155-- Please enter.

Condition Identifier: FDE\$SYSTEM_HELP_MESSAGE Code: FD 155

Description: The terminal user processed a function key for help. The form definition used the default help message.

User Action: To remove the message, enter one of the following: - Type in the data and press the function defined on the form. - Press any key with the cursor on the help form.

--INFORMATIVE FD 156-- Key has no meaning for cursor position.

Condition Identifier: FDE\$SYSTEM_BAD_KEY_MESSAGE Code: FD 156

Description: The terminal user pressed a function key that has no definition on the form where the terminal cursor is positioned.

User Action: Press a function key defined on the form where the cursor is positioned.

--FATAL FL 1-- Value outside of computed GOTO bounds.

Condition Identifier: FLE\$COMPUTED_GOTO Code: FL 1

Description: The value of the expression is less than one or greater than the number of labels in the list. This error occurs in a program that began as a FORTRAN Extended 4 program and was converted to FORTRAN 5 through the F45 conversion aid facility.

User Action: Correct the expression in the computed GO TO statement.

Further Information: Refer to the description of computed GO TO in the FORTRAN Usage Manual.

--FATAL FL 40-- Sense switch number {text} not in range 1..6; return parameter set to 2.

Condition Identifier: FLE\$ILLEGAL_SSWTCH_NUM_ERR Code: FL 40

Description: The first argument in the SSWTCH subroutine call is the number of the sense switch to be tested. The sense switches are numbered from 1 through 6; therefore, the argument must be in this range.

User Action: Specify the first argument as a number in the range 1 through 6.

Further Information: Refer to the description of the SSWTCH subroutine in the FORTRAN Usage Manual.

--FATAL FL 43-- MASK({text}): Argument must be in range 0..64.

Condition Identifier: FLE\$MASK_RANGE Code: FL 43

Description: The argument to the MASK function must be an integer in the range 0 through 64.

User Action: Check the argument of the MASK function for a value less than 0 or greater than 64.

Further Information: Refer to the description of the MASK function in the FORTRAN Usage Manual.

--WARNING FL 51-- SYSTEM utility called with user_nonfatal error {text}: {text}

Condition Identifier: FLE\$USER_NONFATAL Code: FL 51

Description: This message is issued when the SYSTEM routine is called and the first argument is FORTRAN error number 51. (Error #51 is reserved for programmer use.) The second argument to SYSTEM is a message that appears in the system output.

User Action: Depends on why the SYSTEM routine was called.

Further Information: Refer to the description of the SYSTEM call in the FORTRAN usage manual.

--FATAL FL 52-- SYSTEM utility called with user_fatal error {text}: {text}

Condition Identifier: FLE\$USER_FATAL Code: FL 52

Description: This message is issued when the SYSTEM routine is called and the first argument is FORTRAN error number 52. (Error #52 is reserved for programmer use.) The second argument to SYSTEM is a message that appears in the system output.

User Action: Depends on why SYSTEM was called.

Further Information: Refer to the description of the SYSTEM call in the FORTRAN usage manual.

--FATAL FL 55-- Attempt to BUFFERIN past endfile on {file}.

Condition Identifier: FLE\$EOF_ON_BUFFERIN Code: FL 55

Description: The file named in BUFFER IN was positioned at end-of-file when BUFFER IN was executed.

User Action: Rewind the file named in BUFFER IN using REWIND, or insert a call to UNIT to test for end-of-file before BUFFER IN.

Further Information: Refer to the description of BUFFER IN in the FORTRAN Usage Manual.

--FATAL FL 57-- Beginning and ending locations for BUFFER I/O must be in same segment.

Condition Identifier: FLE\$BUFFER_CROSSES_SEGMENT Code: FL 57

Description: Data must be read from or written to one segment; the first and last word addresses must be in the same level of memory.

User Action: Check the first and last locations of the memory block specified in the buffer I/O operation.

Further Information: Refer to the description of BUFFER IN or BUFFER OUT in the FORTRAN Usage Manual.

--FATAL FL 58-- Ending location of BUFFER I/O area precedes beginning location.

Condition Identifier: FLE\$NEG_LENGTH_BUFFER Code: FL 58

Description: The first word address of the buffer I/O area must be less than or equal to the last word address of the buffer I/O area.

User Action: You may have specified the last word of the memory block before the first word, as in the following examples:

```

BUFFERIN(u,p) (A(10), A(1)) <---should be (A(1), A(10))

```

```

COMMON A(10), B(10)

```

```

BUFFERIN(u,p) (B(1),A(10)) <---should be (A(1), B(10))

```

Further Information: Refer to the description of BUFFER IN or BUFFER OUT in the FORTRAN Usage Manual.

--FATAL FL 62-- Unit ({text}) specified does not exist.

Condition Identifier: FLE\$INVALID_UNIT Code: FL 62

Description: The number specified for the UNIT specifier must be an integer in the range 0 through 999 or, if the form L"f" was used, f must be letters or digits beginning with a letter.

User Action: Check the number specified for the UNIT specifier.

Further Information: Refer to the description of the UNIT specifier in the FORTRAN Usage Manual.

--FATAL FL 63-- Attempt to read past endfile on unformatted file {file}.

Condition Identifier: FLE\$EOF_ON_BINARY_READ Code: FL 63

Description: A read operation was attempted on an unformatted file that was positioned at a partition boundary.

User Action: Include the END= specifier on the READ statement to test for end-of-partition. To read past an end-of-partition, you must either perform a CLOSE followed by an OPEN on the file, or specify a call to EOF before reading past the partition.

Further Information: Refer to the description of either OPEN and CLOSE or EOF in the FORTRAN Usage Manual.

--FATAL FL 65-- Attempt to read past endfile on formatted file {file}.

Condition Identifier: FLE\$EOF_ON_CODED_READ Code: FL 65

Description: A read operation was attempted on a formatted file that was positioned at a partition boundary.

User Action: Include the END= specifier on the READ statement to test for end-of-partition. To read past an end-of-partition, you must either perform a CLOSE followed by an OPEN on the file, or specify a call to EOF before reading past the partition.

Further Information: Refer to the description of either OPEN and CLOSE or EOF in the FORTRAN Usage Manual.

--FATAL FL 69-- Initial '(' missing from format spec.

Condition Identifier: FLE\$OPEN_PAREN_MISSING Code: FL 69

Description: A format specification must be bound by parenthesis.

User Action: Ensure that the number of left parentheses in the format specification matches the number of right parentheses.

Further Information: Refer to the description of format specifications in Input/Output chapter in the FORTRAN Usage Manual.

--FATAL FL 70-- Record size of {text} exceeded for formatted I/O on file {file}.

Condition Identifier: FLE\$RECORD_SIZE_EXCEEDED Code: FL 70

Description: A formatted READ or WRITE must not exceed the record length specified for the file.

User Action: Check the format specification or the I/O list to ensure that the specified format corresponds to the format of the file.

Further Information: Refer to the description of input/output lists in the Input/Output chapter of the FORTRAN Usage Manual.

--FATAL FL 71-- Width field of edit descriptor cannot be 0.

Condition Identifier: FLE\$ZERO_W_FIELD Code: FL 71

Description: The field width of an edit descriptor must be greater than zero.

User Action: Change the field width of the edit descriptor to a positive integer.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 72-- Number of fractional digits ({text}) must be < field width ({text}).

Condition Identifier: FLE\$W_LE_D Code: FL 72

Description: In the w.d form of an edit descriptor, w specifies the field width, including sign and decimal point, and d specifies the number of digits to the right of the decimal point. w must be less than or equal to d.

User Action: Change the edit descriptor so that d is less than or equal to w.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 73-- No repeatable edit descriptor in present format group.

Condition Identifier: FLE\$NO_REPEAT_DESCRIPTOR Code: FL 73

Description: Every data item in the iolist of a READ or WRITE must be described by a repeatable edit descriptor in a format specification.

User Action: Change the FORMAT specification to specify a repeatable edit descriptor for each data item to be read or written.

Further Information: Refer to the description of format specifications in the FORTRAN Usage Manual.

--FATAL FL 78-- Data type mismatch. Input type is {text}; list type is {text}.

Condition Identifier: FLE\$DATA_TYPE_MISMATCH_NAMIN Code: FL 78

Description: The data types of a variable and its assigned data must match. For example, a character variable must be assigned character data, a logical variable must be assigned logical data, and so forth.

User Action: Change the variable or its data to match data types.

Further Information: Refer to the description of namelist input/output in the FORTRAN Usage Manual.

--FATAL FL 79-- Input data value outside of machine range.

Condition Identifier: FLE\$DATA_OVERFLOW Code: FL 79

Description: A real, double precision, complex, or integer input value must not exceed the maximum value allowed by the computer.

User Action: Change the input values so that they are within the limits allowed by the computer.

Further Information: Refer to the description of valid ranges for numeric values in Chapter 2 of the FORTRAN Usage Manual.

--FATAL FL 83-- Print limit of {text} lines for \$OUTPUT exceeded.

Condition Identifier: FLE\$PRINT_LIMIT_EXCEEDED Code: FL 83

Description: The number of records written to file \$OUTPUT or \$ERRORS has exceeded either the default value (5000 lines), or the limit specified by the \$PRINT_LIMIT parameter on the program execution (LGO) command.

User Action: This error usually indicates the presence of an infinite loop; if so, correct the loop. If not, increase the value of the maximum number of records output to files \$OUTPUT and \$ERRORS using the \$PRINT_LIMIT parameter on the program execution command.

Further Information: Refer to the description of the \$PRINT_LIMIT parameter in the FORTRAN Usage Manual.

--FATAL FL 88-- Read attempted after write on {file}.

Condition Identifier: FLE\$INPUT_AFTER_OUTPUT Code: FL 88

Description: A sequential file that has been written to must be repositioned before it can be read by a READ or BUFFER IN.

User Action: Reposition the file using REWIND or BACKSPACE before the READ or BUFFER IN.

Further Information: Refer to the description of REWIND or BACKSPACE in the FORTRAN Usage Manual.

--FATAL FL 89-- Insufficient data in record input from {file} to satisfy input list.

Condition Identifier: FLE\$LIST_EXCEEDS_DATA Code: FL 89

Description: The number of data items in the input record is less than the number of items requested on the READ.

User Action: Possible causes are an incorrect unit (or file associated with the unit) on the READ, a record with too few data items, or a READ with too many data items. Check the file name, the file contents, or the READ.

Further Information: Refer to the description of input/output lists in the FORTRAN Usage Manual.

--FATAL FL 90-- Parity error on {file} during previous read.

Condition Identifier: FLE\$AFTER_PARITY_ERR_INPB Code: FL 90

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 92-- Parity error on {file} during previous read.

Condition Identifier: FLE\$AFTER_PARITY_ERR_INPC Code: FL 92

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 93-- Parity error on {file} during previous write.

Condition Identifier: FLE\$AFTER_PARITY_ERR_OUTB Code: FL 93

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 94-- Parity error on {file} during previous write.

Condition Identifier: FLE\$AFTER_PARITY_ERR_OUTC Code: FL 94

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 95-- Parity error on {file} during previous write.

Condition Identifier: FLE\$AFTER_PARITY_ERR_ODA Code: FL 95

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 96-- Parity error on {file} during previous read.

Condition Identifier: FLE\$AFTER_PARITY_ERR_IDA Code: FL 96

Description: Probable disk or tape error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 97-- Mass storage number key must be in range 1..{text}.

Condition Identifier: FLE\$MS_NUMBER_KEY_INVALID Code: FL 97

Description: A number key must be less than the length (in words) of the index.

User Action: Check the index length specified in the OPENMS or STINDEX call and change all number keys to fall within the required range.

Further Information: Refer to the description of READMS or WRITMS in the FORTRAN Usage Manual.

--FATAL FL 98-- Mass storage I/O attempted on unopened unit {text}.

Condition Identifier: FLE\$MS_UNIT_NOT_OPEN Code: FL 98

Description: A READMS or WRITMS operation cannot be performed on an unopened I/O unit.

User Action: Specify the unit in an OPENMS call before the READMS or WRITMS.

Further Information: Refer to the description of OPENMS in the FORTRAN Usage Manual.

--FATAL FL 99-- Master index of {file} has {text} keys. OPENMS specified {text} keys.

Condition Identifier: FLE\$MS_INDEX_TYPE_MISMATCH Code: FL 99

Description: The key type specified in the OPENMS call does not match the key type specified when the file was created.

User Action: Ensure that you have specified the correct file name in the OPENMS call, then change the key type specified in the OPENMS call to the type specified at creation.

Further Information: Refer to the description of OPENMS in the FORTRAN Usage Manual.

--FATAL FL 100-- Index for {file} is full. It contains {text} entries.

Condition Identifier: FLE\$MS_INDEX_FULL Code: FL 100

Description: The mass storage unit index is full; therefore, no more records can be added.

User Action: Extend the size of the index by increasing the number of elements in the index array on the DIMENSION statement.

Further Information: Refer to the description of mass storage I/O in the FORTRAN Usage Manual.

--FATAL FL 101-- File {file} not in mass storage I/O format.

Condition Identifier: FLE\$MS_DEFECTIVE_FILE Code: FL 101

Description: The file specified in a mass storage I/O operation (OPENMS, READMS, or WRITMS) is not a mass storage file.

User Action: Change the unit (or file name associated with the unit) specified on the mass storage operation statement. In order to be used in a mass storage operation, an existing file must have been created as a mass storage file (initially opened as a mass storage file).

Further Information: Refer to the description of mass storage I/O in the FORTRAN Usage Manual.

--FATAL FL 102-- Input buffer for {file} must be \geq {text}.

Condition Identifier: FLE\$BUFFER_TOO_SMALL Code: FL 102

Description: A BUFFER IN transferred a type FIXED record that was longer than the buffer length specified.

User Action: Increase the size of the memory block specified on the BUFFER IN.

Further Information: Refer to the description of BUFFER IN in the FORTRAN Usage Manual.

--FATAL FL 103-- Abnormal status condition {text} returned from system routine.

Condition Identifier: FLE\$SYSTEM_RETURNED Code: FL 103

Description: An error occurred in a system routine. The error message is generally followed by another more descriptive message issued by FORTRAN.

User Action: Correct the error indicated in the FORTRAN message.

--FATAL FL 104-- Record with {text} key {text} not found on {file}.

Condition Identifier: FLE\$MS_RECORD_NOT_FOUND Code: FL 104

Description: The key that was referenced in a mass storage operation does not exist in the file index.

User Action: Check for an incorrect unit (or file name associated with the unit) or incorrect record key specification in the mass storage I/O operation.

Further Information: Refer to the description of mass storage I/O in the FORTRAN Usage Manual.

--FATAL FL 105-- Attempted WRITMS of record with length ≤ 0 to {file}.

Condition Identifier: FLE\$MS_RECORD_LENGTH_NEGATIVE Code: FL 105

Description: The record length specified for a mass storage file must not be negative or zero.

User Action: Ensure that the record length specification on the WRITMS is a positive integer.

Further Information: Refer to the description of mass storage I/O in the FORTRAN Usage Manual.

--FATAL FL 106-- Attempted rewrite of record larger than existing record on {file} at key {text}.

Condition Identifier: FLE\$MS_RECORD_TOO_LONG Code: FL 106

Description: The length of a new record that is to replace an existing record must not exceed the length of the existing record.

User Action: Check the record length specified in the WRITMS call. If it is correct, either reduce the length of the new record or specify 0 or -1 for the rewrite parameter on the WRITMS call, so that the oversized record is written after the last record on the file.

Further Information: Refer to the description of mass storage I/O in the FORTRAN Usage Manual.

--FATAL FL 114-- Invalid option was specified for CALL CONNEC.

Condition Identifier: FLE\$INVALID_OPTION_FOR_CONNEC Code: FL 114

Description: The call to CONNEC has an incorrect argument. The format of the CONNEC call is

CALL CONNEC (u,cs)

where u is the input/output unit (usually an integer) and cs selects a character set (integer 1, 2, or 3).

User Action: Specify correct arguments in the CONNEC call.

Further Information: Refer to the description of the CONNEC subroutine call under Input/Output Related Statements in the FORTRAN usage manual.

--FATAL FL 132-- Duplicate character in CSOWN argument at position {text}.

Condition Identifier: FLE\$CSOWN_DUPL_CHAR Code: FL 132

Description: The argument specified in the CSOWN call must not contain duplicate characters.

User Action: Change the argument so that no duplicate characters are specified.

Further Information: Refer to the description of the CSOWN subroutine in the FORTRAN Usage Manual.

--FATAL FL 133-- FILE= spec must not be specified on an OPEN with STATUS='SCRATCH'.

Condition Identifier: FLE\$OPEN_SCR_WITH_FILE_NAME Code: FL 133

Description: In ANSI FORTRAN, FILE=spec cannot be specified on an OPEN with STATUS='SCRATCH'.

User Action: Remove either the FILE=filename or the STATUS='SCRATCH' specification from the OPEN.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 134-- STATUS={text} must not be specified for a file that {text}.

Condition Identifier: FLE\$INCONSISTANT_STATUS_OPTION Code: FL 134

Description: The option specified for the STATUS parameter on the OPEN statement must be consistent with the status of the file.

User Action: Specify a STATUS=option that is consistent with the status of the file, specify STATUS='UNKNOWN', or omit the STATUS parameter.

Further Information: Refer to the description of the OPEN statement format in the FORTRAN Usage Manual.

--FATAL FL 135-- Value of {text} spec does not match previous OPEN of {text}.

Condition Identifier: FLE\$REDUN_OPEN_OPTION_DIFFERS Code: FL 135

Description: The indicated specification on an OPEN of a file must match the same specification on the first OPEN of the file.

User Action: Check the parameters on the two OPENS and change one set to match the other, or change the unit (or file associated with the unit) on one of the OPENS, or CLOSE the file before reopening it.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 136-- RECL must be > 0.

Condition Identifier: FLE\$RECL_NOT_POSITIVE Code: FL 136

Description: The value specified for the record length of a file must be greater than zero.

User Action: Specify a nonzero positive integer for the record length of the file.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 137-- BLANK= spec must not be used on an OPEN of an unformatted file.

Condition Identifier: FLE\$OPEN_UNFMTD_FILE_WITH_BLANK Code: FL 137

Description: The BLANK= specifier must not be used on an unformatted file because blank conversion is not meaningful for unformatted I/O.

User Action: Remove the BLANK= specification from the OPEN, or change the formatting option to FORMATTED.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 138-- BUFL must be ≥ 0 .

Condition Identifier: FLE\$BUFL_NEGATIVE Code: FL 138

Description: The value of the BUFL parameter cannot be less than 0. For files that have been previously opened, the value of the BUFL parameter must equal the file buffer length. Although the BUFL parameter is not used under NOS/VE, this message is issued to maintain compatibility with NOS.

User Action: Remove the BUFL parameter or, if your application is also used under NOS, change the value of the BUFL parameter to a positive integer.

Further Information: Refer to the description of the BUFL parameter in the FORTRAN Version 5 Reference Manual.

--FATAL FL 141-- Argument of ICHAR must be a string of length 1.

Condition Identifier: FLE\$ICHAR_ARG_TOO_LONG Code: FL 141

Description: The argument of the ICHAR function must be either a single character or a character expression whose value is a single character.

User Action: Change the function argument to a character string of length one.

Further Information: Refer to the description of the ICHAR function in the FORTRAN Usage Manual.

--FATAL FL 144-- {text} {text} of {text}: Subscript {text} of array {text} is {text}. Declared lower bound is {text}; upper bound is {text}.

Condition Identifier: FLE\$SUBSCRIPT_ERROR Code: FL 144

Description: The array subscript is not within the declared array bounds.

User Action: Check program logic for errors that would cause the subscript to be out-of-bounds.

Further Information: Refer to the description of array storage in the FORTRAN Usage Manual.

--FATAL FL 145-- {text} {text} of {text}: Beginning character position ({text}) must be > 0 .

Condition Identifier: FLE\$SUBSTR_BEGIN_TOO_SMALL Code: FL 145

Description: String elements are numbered from 1 through n, where n is the position of the last element in the string. A substring cannot begin before the first element in a string; therefore, the beginning character position in a substring reference must be greater than zero.

User Action: Change the substring expression so that the value of the reference to the first character position is $1 \leq \text{substring expression} \leq n$.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 146-- {text} {text} of {text}: Ending character position ({text}) must be .GE. beginning character position ({text}).

Condition Identifier: FLE\$SUBSTR_END_TOO_SMALL Code: FL 146

Description: The value of the expression that references the last position of a substring must be greater than the value of the expression that references the first position.

User Action: Change the substring expression so that the value of the reference to the last character position is greater than or equal to the value of the reference to the beginning character position.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 147-- {text} {text} of {text}: Ending character position ({text}) must be .LE. string length ({text}).

Condition Identifier: FLE\$SUBSTR_END_TOO_BIG Code: FL 147

Description: The value of the expression that references the last character in a substring must be less than or equal to the length of the string.

User Action: Specify an expression whose value is less than or equal to the last character position of the string.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 150-- Iolist exceeds internal file.

Condition Identifier: FLE\$IO_LIST_EXCEEDS_INT_FILE Code: FL 150

Description: The number of items in the READ input list must not be greater than the number of items that are specified by the format specification in the internal file.

User Action: Reduce the number of items in the READ input list, or check the format specification for field widths that are too large.

Further Information: Refer to the description of internal files in the FORTRAN Usage Manual.

--FATAL FL 151-- RECL must be specified for direct access OPEN (unit {text}).

Condition Identifier: FLE\$OPEN_DIRECT_WITHOUT_RECL Code: FL 151

Description: The record length parameter must be specified in the OPEN of a direct access file.

User Action: Specify the RECL parameter in the OPEN of the file.

Further Information: Refer to the descriptions of OPEN and direct access file usage in the FORTRAN Usage Manual.

--FATAL FL 152-- REWIND not permitted on direct access file ({file}).

Condition Identifier: FLE\$REWIND Code: FL 152

Description: It is not necessary to position a direct access file. Direct access files read or write a record directly, without searching sequentially from a specific position.

User Action: Be sure that the correct unit was specified in the REWIND, or remove the REWIND.

Further Information: Refer to the description of direct access files in the FORTRAN Usage Manual.

--FATAL FL 153-- Argument of CSOWN must be of type CHARACTER.

Condition Identifier: FLE\$CSOWN_ARG_NOT_CHAR Code: FL 153

Description: The argument of the CSOWN function must be of type CHARACTER.

User Action: Be sure that you have specified the correct argument and that it is declared in a CHARACTER statement.

Further Information: Refer to the description of the CSOWN function in the FORTRAN Usage Manual.

--FATAL FL 155-- {file} requires {text} access.

Condition Identifier: FLE\$USAGE_MODE_SEQ Code: FL 155

Description: The access method of the named file is not compatible with the type of operation you requested.

User Action: Check the ACCESS= specifier on the OPEN of the file for incorrect access method. (Omission of the ACCESS= parameter implies sequential access.) Be sure that you are performing the correct type of input/output on the file.

Further Information: Refer to the description of the OPEN statement and of the file access methods in the FORTRAN Usage Manual.

--FATAL FL 157-- Unknown collating sequence: {text}.

Condition Identifier: FLE\$UNKNOWN_COLSEQ_NAME Code: FL 157

Description: The specified collating sequence is not recognizable to FORTRAN.

User Action: Change the collating sequence specification to one of the following strings: 'ASCII', 'ASCII6', 'ASCII6S', 'COBOL6', 'COBOL6S', 'DISPLAY', 'DISPLAYS', 'DISPLAY63', 'DISPLAY63S', 'EBCDIC', 'EBCDIC6', 'EBCDIC6S', 'INSTALL'

Further Information: Refer to the description of the COLSEQ routine in the FORTRAN Usage Manual.

--FATAL FL 158-- Insufficient free space left to process a repeated string.

Condition Identifier: FLE\$NO_SPACE_FOR_REPTD_STR Code: FL 158

Description: An internal routine exceeded the bounds of a segment (2**31 bytes) while transforming a repeated string to an internal representation.

User Action: Follow site-defined procedures for reporting system errors. This error can be caused by an infinite loop in your program, an incorrect record length specification on an OPEN, or an error in the system software.

--FATAL FL 159-- Scratch file on unit {text} must not be closed with STATUS='KEEP'.

Condition Identifier: FLE\$CLOSE_KEEP_ON_SCRATCH_FILE Code: FL 159

Description: STATUS='KEEP' must not be specified on the CLOSE of a file that was opened with STATUS='SCRATCH'.

User Action: Either change the STATUS= specifier on the OPEN or remove the STATUS= specifier on the CLOSE and let the file be deleted.

Further Information: Refer to the description of either OPEN or CLOSE in the FORTRAN Usage Manual.

--FATAL FL 167-- Hollerith edit descriptors must not be used for input.

Condition Identifier: FLE\$H_FORMAT_ON_INPUT Code: FL 167

Description: The nH edit descriptor is used to output character strings, and cannot be used for input.

User Action: Remove the nH descriptor from the format specification.

Further Information: Refer to the description of the nH descriptor in the FORTRAN Usage Manual.

--FATAL FL 168-- Decimal point missing in edit descriptor.

Condition Identifier: FLE\$DECIMAL_MISSING_IN_FORMAT Code: FL 168

Description: This edit descriptor requires a decimal point between the number that specifies the total width of the field and the number that specifies the number of digits that are to be displayed to the right of the decimal point.

User Action: Insert a decimal point in the appropriate place in the edit descriptor.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 169-- Format variable must contain an assigned format label.

Condition Identifier: FLE\$NOT_ASSIGNED_FMT Code: FL 169

Description: The variable specified in the FMT= specifier does not contain the label of a FORMAT statement.

User Action: Use an ASSIGN statement to assign a FORMAT label to the variable.

Further Information: Refer to the description of ASSIGN in the FORTRAN Usage Manual.

--FATAL FL 200-- OPEN must be done on unit {text} before direct access write.

Condition Identifier: FLE\$WRITE_ON_UNOPENED_FILE Code: FL 200

Description: A file that has not been opened was specified in a WRITE.

User Action: Either open the file before writing to it, or check for an incorrect unit number in the WRITE statement.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 201-- {file} requires {text} access.

Condition Identifier: FLE\$USAGE_MODE_DIR_ODAB Code: FL 201

Description: An unformatted direct access WRITE was performed on a file that was opened for sequential or random access.

User Action: Be sure that ACCESS='DIRECT' is specified on the OPEN statement. Check for incorrect unit number on the READ or WRITE statement.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 202-- {file} requires formatted I/O.

Condition Identifier: FLE\$BINARY_ON_CODED_OUT Code: FL 202

Description: An unformatted I/O operation cannot be performed on a direct access file that was opened for formatted I/O operations.

User Action: Be sure that the correct unit was specified in the OPEN of the file. If the unit is correct, change the I/O operation to a formatted I/O operation.

Further Information: Refer to the description of direct access or formatted input/output or of the OPEN Statement in the FORTRAN Usage Manual.

--FATAL FL 203-- REC must be > 0. Value is {text}.

Condition Identifier: FLE\$REC_NOT_POSITIVE_ODA Code: FL 203

Description: The RECL specifier in a direct access WRITE operation cannot specify a value that is less than or equal to zero.

User Action: Check the value of the RECL specifier in the OPEN of the file.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 204-- OPEN must be done on unit {text} before direct access read.

Condition Identifier: FLE\$READ_ON_UNOPENED_FILE Code: FL 204

Description: A file that was not previously opened was specified in a READ operation.

User Action: Be sure that you open the file before reading it, and that you have specified the correct unit number in the READ statement.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 205-- {file} requires {text} access.

Condition Identifier: FLE\$USAGE_MODE_DIR_IDAB Code: FL 205

Description: An unformatted direct access READ cannot be performed on a file that was opened for sequential or random access.

User Action: Specify ACCESS='DIRECT' on the OPEN of the file.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 206-- {file} requires formatted I/O.

Condition Identifier: FLE\$BINARY_ON_CODED_IN Code: FL 206

Description: An unformatted I/O operation cannot be performed on a direct access file that was opened for formatted I/O.

User Action: Specify FORM='UNFORMATTED' on the OPEN of the file.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 207-- REC must be > 0. Value is {text}.

Condition Identifier: FLE\$REC_NOT_POSITIVE_IDA Code: FL 207

Description: The value passed by the REC= specifier in a direct access READ statement must be a positive integer.

User Action: Change the value of the REC= specifier to a positive integer.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 208-- Iolist exceeds RECL={text} for {file}.

Condition Identifier: FLE\$EXCESS_DATA_ODA Code: FL 208

Description: The length of the record written by a direct access WRITE exceeded the length specified by the RECL= specifier on the OPEN of the direct access file.

User Action: Either specify fewer data items in the WRITE statement or increase the value of the RECL= specifier in the OPEN statement.

Further Information: Refer to the description of either WRITE or OPEN in the FORTRAN Usage Manual.

--FATAL FL 210-- {file} requires {text} access.

Condition Identifier: FLE\$USAGE_MODE_DIR_ODAC Code: FL 210

Description: A formatted direct access WRITE was performed on a file that was opened for sequential or random access.

User Action: Either change the OPEN statement to specify ACCESS='DIRECT' or change the I/O statement to a sequential access or a random access statement.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 211-- {file} requires unformatted I/O.

Condition Identifier: FLE\$CODED_ON_BINARY_FILE Code: FL 211

Description: A formatted I/O operation was attempted on a direct access file that contains unformatted data.

User Action: Be sure you have specified the correct unit number in the I/O statement. Then change the I/O operation to an unformatted operation (that is, one with the FMT= specifier omitted).

Further Information: Refer to the description of unformatted input/output in the FORTRAN Usage Manual.

--FATAL FL 213-- {file} requires {text} access.

Condition Identifier: FLE\$USAGE_MODE_DIR_IDAC Code: FL 213

Description: A direct access READ was performed on a file that was created as a sequential or random access file.

User Action: Change the READ statement to match the file access type.

Further Information: Refer to the discussion of READ statements and file access methods in the FORTRAN Usage Manual.

--FATAL FL 215-- Undefined weight of {text} passed to CHAR.

Condition Identifier: FLE\$CHAR_WEIGHT_UNDEFINED Code: FL 215

Description: The character weight assigned to the CHAR argument either is negative or exceeds the character weight assigned to the last character in the collating sequence.

User Action: Ensure that the correct collating sequence was specified (default is ASCII), or check for an error that caused an invalid weight to be passed.

Further Information: Refer to the description of intrinsic functions or collating sequence control in the FORTRAN Usage Manual.

--FATAL FL 216-- Colon missing from substring reference descriptor.

Condition Identifier: FLE\$COLON_MISSING_SUBSTRING Code: FL 216

Description: A colon is missing from a substring reference.

User Action: Insert a colon in the appropriate place in the substring reference.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 217-- Upper bound of {text} exceeds string length of {text}.

Condition Identifier: FLE\$SUBSTRING_UPPER_BOUND Code: FL 217

Description: The substring reference specifies a character beyond the length of the named character string.

User Action: Ensure that the correct character string was named in the substring reference, or decrease the upper bound of the substring reference.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 218-- Number of units requested exceeds max number allowed ({text}).

Condition Identifier: FLE\$TOO_MANY_UNITS Code: FL 218

Description: The maximum number of units allowed to be open at one time was exceeded.

User Action: Close some units using CLOSE, or check for an infinite loop that opens units.

Further Information: Refer to the description of CLOSE in the FORTRAN Usage Manual.

--FATAL FL 219-- Insufficient space exists to open another unit.

Condition Identifier: FLE\$UNIT_TABLE_OVERFLOW Code: FL 219

Description: The maximum number of units that can be open was exceeded. This is a possible internal error.

User Action: Follow site-defined procedures for reporting software errors.

**--FATAL FL 220-- Number of files requested exceeds max number allowed
(`{text}`).**

Condition Identifier: FLE\$TOO_MANY_FILES Code: FL 220

Description: The number of files requested to be open exceeded the maximum number allowed.

User Action: Close some files using CLOSE statements.

Further Information: Refer to the description of CLOSE in the FORTRAN Usage Manual.

--FATAL FL 221-- Insufficient space exists to open another file.

Condition Identifier: FLE\$FILE_TABLE_OVERFLOW Code: FL 221

Description: There is not enough internal table space to open another file.

User Action: Close some files using CLOSE, or check for an infinite loop that opens files.

Further Information: Refer to the description of CLOSE in the FORTRAN Usage Manual.

--FATAL FL 222-- Insufficient space exists to allocate a record of length `{text}`.

Condition Identifier: FLE\$WSA_OVERFLOW Code: FL 222

Description: There is not enough space in the internal working storage area to add another record to the file.

User Action: Check for an infinite loop that writes records to the file. Check the RECL= specifier in the OPEN statement for an invalid value.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 223-- Insufficient space exists to process `{text}` SCL parameters.

Condition Identifier: FLE\$PDT_OVERFLOW Code: FL 223

Description: This is a possible internal software error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 224-- Internal file array has a dimension < 1.

Condition Identifier: FLE\$INT_FILE_SIZE_NOT_POSITIVE Code: FL 224

Description: The dimension of an array used as an internal file must be 1 or greater.

User Action: Check for a dimension specification in which the lower bound is equal to or greater than the upper bound. If the array has adjustable dimensions (dimension bounds passed through an argument list), check for the following:

```
SUBROUTINE SUB (A, M, N)
```

```
    DIMENSION ARRAY A(N:M)           where (M-N+1) < 0
```

This error could also be caused by the same problem with the dimension statement of an array in COMMON.

Further Information: Refer to the description of adjustable dimensions in the FORTRAN Usage Manual.

--FATAL FL 225-- Format arrays must have < `{text}` elements.

Condition Identifier: FLE\$FMT_SIZE_TOO_BIG Code: FL 225

Description: A format specification array exceeded the maximum allowed number of elements.

User Action: Check the length of the array format and shorten it if necessary. A likely cause of this error is an uninitialized array dimension variable.

--FATAL FL 226-- Format is larger than a segment.

Condition Identifier: FLE\$FMT_OVERFLOW_SEGMENT Code: FL 226

Description: The number of characters in a format specification exceeds the bounds of a segment.

User Action: Check for an excessively large format specification. A possible cause is an improperly dimensioned array that contains a format specification.

Further Information: Refer to the description of format specifications in the FORTRAN Usage Manual.

--FATAL FL 227-- RECL for OPEN must be < {text} characters ({text} words).

Condition Identifier: FLE\$RECL_TOO_BIG Code: FL 227

Description: The value specified for the RECL= specifier in an OPEN statement was too large.

User Action: Check the value of the RECL specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 228-- FILE= spec does not contain a valid file name.

Condition Identifier: FLE\$INVALID_FILE_NAME Code: FL 228

Description: The file name specified is not a valid NOS/VE file name.

User Action: Correct the file name. The name must be from 1 to 31 characters chosen from A-Z, a-z, 0-9, @, ., #, and \$; the first character cannot be a digit.

Further Information: Refer to Introduction to Input/Output in the FORTRAN Usage Manual.

--FATAL FL 229-- FORM must be 'FORMATTED' or 'UNFORMATTED' or 'BUFFERED'.

Condition Identifier: FLE\$INVALID_FORM_SPEC Code: FL 229

Description: The FORM= specifier in the OPEN statement must specify a character expression whose value is either 'FORMATTED' or 'UNFORMATTED'.

User Action: Correct the value specified for the FORM= specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 230-- ACCESS must be 'SEQUENTIAL' or 'DIRECT'.

Condition Identifier: FLE\$INVALID_ACCESS_SPEC Code: FL 230

Description: The ACCESS= specifier in the OPEN statement must specify a character expression whose value is either 'SEQUENTIAL' or 'DIRECT'.

User Action: Correct the value specified for the ACCESS= specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 231-- BLANK must be 'NULL' or 'ZERO'.

Condition Identifier: FLE\$INVALID_BLANK_SPEC Code: FL 231

Description: The BLANK= specifier in the OPEN statement must specify a character expression whose value is either 'NULL' or 'ZERO'.

User Action: Correct the value specified for the BLANK= specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 232-- BUFL= spec requests more than a segment.

Condition Identifier: FLE\$BUFL_BIGGER_THAN_SEGMENT Code: FL 232

Description: The value of the BUFL= specifier was too large (exceeded 2**31 bytes).

User Action: Check the value of the BUFL= specifier and reduce the value if necessary.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 233-- REC must be < {text}.

Condition Identifier: FLE\$REC_EXCEEDS_MAX_FILE_LEN Code: FL 233

Description: The record number specified for the REC= specifier in a direct access READ or WRITE statement is beyond the maximum number of records allowed on a file.

User Action: Check the REC= specifier for an excessively large value.

--FATAL FL 234-- STATUS must be 'OLD', 'NEW', 'SCRATCH' or 'UNKNOWN'.

Condition Identifier: FLE\$INVALID_OPEN_STATUS_SPEC Code: FL 234

Description: The value of the STATUS= specifier in an OPEN statement must be either 'OLD', 'SCRATCH', or 'DELETE'.

User Action: Correct the value of the STATUS= specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 235-- STATUS must be 'KEEP' or 'DELETE'.

Condition Identifier: FLE\$INVALID_CLOSE_STATUS_SPEC Code: FL 235

Description: The value of the STATUS= specifier in a CLOSE statement must be either 'KEEP' or 'DELETE'.

User Action: Correct the value of the STATUS= specifier.

Further Information: Refer to the description of CLOSE in the FORTRAN Usage Manual.

--FATAL FL 236-- Format array has a dimension < 1.

Condition Identifier: FLE\$FMT_SIZE_NOT_POSITIVE Code: FL 236

Description: The dimension bounds of an adjustable dimension array have a negative or zero value. For example:

```
SUBROUTINE SUB (A, N, M)

  DIMENSION A(N:M)   where (M-N+1) < 0.
```

or

```
DIMENSION A(N*M)   where (N*M) <= 0.
```

This error could also be caused by the same problem with the dimension statement of an array in COMMON.

User Action: Check the array declaration for reversed or mistyped dimensions.

Further Information: Refer to the description of adjustable dimensions in the FORTRAN Usage Manual.

--FATAL FL 237-- An internal file must contain < {text} records.

Condition Identifier: FLE\$INT_FILE_SIZE_TOO_BIG Code: FL 237

Description: The dimensions of an array used as an internal file exceed the maximum number of records allowed for an internal file.

User Action: Check the dimension bounds of the array for invalid values, or reduce the number of array elements to less than 16,777,216.

--FATAL FL 238-- Direct access file {file} must have record type F.

Condition Identifier: FLE\$DIR_ACC_REC_TYPE Code: FL 238

Description: The indicated file does not have the correct record type to be opened as a direct access file.

User Action: Either specify a file having the correct record type (FIXED-type records) or perform a different type of I/O on the file. It is possible to change the record type of a file, but it is not advisable because of the possibility of destroying the contents of the file.

Further Information: Refer to the description of Input/Output Implementation in the FORTRAN Usage Manual.

--FATAL FL 239-- Mass storage I/O file ({file}) must have record type U.

Condition Identifier: FLE\$MSIO_REC_TYPE Code: FL 239

Description: The indicated file does not have the correct record type (UNDEFINED-type) to be opened as a mass storage I/O file.

User Action: different type of I/O on the file.

Further Information: Refer to the description of I/O Implementation files in the FORTRAN Usage Manual.

--FATAL FL 240-- Direct access file ({file}) must have byte addressable file organization.

Condition Identifier: FLE\$NOT_BYTE_ADDRESSABLE_DIR Code: FL 240

Description: The file you tried to open for direct access does not have the required organization. If an existing file is to be opened for direct access, it must have been created as a direct access file.

User Action: You may have specified the wrong file in the OPEN statement. Either specify a file that has the correct organization or change the access method to one that is valid for the existing file.

Further Information: Refer to the description of I/O Implementation in the FORTRAN Usage manual.

--FATAL FL 241-- Mass storage file ({file}) must have byte addressable file organization.

Condition Identifier: FLE\$NOT_BYTE_ADDRESSABLE_RAN Code: FL 241

Description: The file you specified in the OPENMS statement does not have the required organization. If a file is to be opened for mass storage I/O, it must have been created as a mass storage file.

User Action: Either specify a file having the correct organization or change the method of input/output to one that is valid for the file.

Further Information: Refer to the descriptions of mass storage input/output and I/O implementation in the FORTRAN Usage Manual.

--FATAL FL 242-- RECL for {file} must be {text} bytes.

Condition Identifier: FLE\$RECL_NOT_SAME_AS_MRL Code: FL 242

Description: On the OPEN of an existing direct access file, the specified record length (RECL= specifier) must match the value specified when the file was created.

User Action: Either omit the RECL= specifier (it defaults to the existing value) or change it to specify the correct record length.

Further Information: Refer to the description of direct access files in the FORTRAN Usage Manual.

--FATAL FL 243-- UNIT must be called after a BUFFER I/O operation (file {file}).

Condition Identifier: FLE\$BUSY Code: FL 243

Description: After a buffer I/O operation, the UNIT function must be called before a subsequent operation can be performed on the same file.

User Action: Insert a call to the UNIT function after the BUFFER statement.

Further Information: Refer to the description of BUFFER IN, BUFFER OUT, or the UNIT function in the FORTRAN Usage Manual.

--FATAL FL 244-- Output buffer length for {file} must be <= {text}.

Condition Identifier: FLE\$BUFFER_TOO_LARGE Code: FL 244

Description: The value of the BUFL= specifier on the OPEN of the file exceeds the maximum allowed buffer length (2**42 bytes).

User Action: Check the value of the BUFL= specifier.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 245-- Character not expected in this position in list-directed input.

Condition Identifier: FLE\$BAD_CHAR_LIST_IN Code: FL 245

Description: A constant (digit, ', ") was found in a character string in list-directed input.

User Action: Either insert a comma or blank to separate the constant from the character string or remove the constant from the character string.

Further Information: Refer to the description of list-directed input in the FORTRAN Usage Manual.

--FATAL FL 246-- Exponent expected following E or D.

Condition Identifier: FLE\$EXPONENT_MISSING Code: FL 246

Description: A single or double precision number must have an exponent following the E or D.

User Action: Specify an exponent after the E or D.

Further Information: Refer to the description of list directed input/output in the FORTRAN Usage Manual.

--FATAL FL 247-- Contiguous repeat counts are not accepted.

Condition Identifier: FLE\$CONTIGUOUS_REPEAT_COUNTS Code: FL 247

Description: A repeat count must be followed by either an asterisk (2*) or an asterisk and a value to be repeated (2*3); it cannot be followed by an asterisk and another repeat count (2*3*4.5).

User Action: Either combine the two repeat counts (6*4.5) or insert a comma to separate the expression into two repeat count/value expressions (2*, 3*4.5).

Further Information: Refer to the description of repeat counts under List-Directed Input in the FORTRAN Usage Manual.

--FATAL FL 248-- Repeat count must be an unsigned integer.

Condition Identifier: FLE\$REPEAT_COUNT_FLOAT_OR_SIGN Code: FL 248

Description: A signed integer was found preceding an asterisk in the list directed input record. The asterisk indicates that the integer is a repeat count; however, a repeat count must be an unsigned integer.

User Action: Remove the minus sign from the repeat count.

Further Information: Refer to the description of repeat counts under List Directed Input in the FORTRAN Usage Manual.

--FATAL FL 249-- Repeat count is invalid.

Condition Identifier: FLE\$INVALID_REPEAT_COUNT Code: FL 249

Description: The integer acting as a repeat count (preceding the asterisk) in the list directed input record is larger than the largest integer allowed on the computer ((2**64)-1).

User Action: Check for an invalid repeat count or a missing comma or blank.

Further Information: Refer to the description of repeat counts under List Directed Input in the FORTRAN Usage Manual.

--FATAL FL 250-- Data type mismatch. Input type is {text}; list type is {text}.

Condition Identifier: FLE\$DATA_TYPE_MISMATCH Code: FL 250

Description: The type of the item read from list directed input must match the type of the item as declared in the program.

User Action: Check for an error in either the type of the item read from list directed input or the type of the item declaration in the program.

Further Information: Refer to the description of list directed input in the FORTRAN Usage Manual.

--FATAL FL 251-- Error in Boolean constant.

Condition Identifier: FLE\$BOOLEAN_CONST_ERROR Code: FL 251

Description: An O or Z in list directed input, which signals the start of a boolean constant, must be followed by a quote (") symbol. Another cause of this error is a non-hexadecimal or non-octal digit found in the boolean constant.

User Action: Either add a quote (") symbol to the beginning of the boolean constant or remove the non-hexadecimal or non-octal digit from the boolean constant.

Further Information: Refer to the description of list directed input/output in the FORTRAN Usage Manual.

--FATAL FL 252-- Internal file record length must be > 0.

Condition Identifier: FLE\$INT_FILE_RECL_TOO_SMALL Code: FL 252

Description: A dimension bound of an internal file array declaration has a negative value.

User Action: In the array declaration for the internal file, check for an inverted array bounds definition, for an uninitialized array bound variable, or a situation similar to the following:

```
SUBROUTINE SUB (A, M, N)
```

```
  DIMENSION A(M,N)   where (N-M+1) <= 0
```

or

```
  DIMENSION A(M,N)   where (M*N) <= 0.
```

Further Information: Refer to the description of internal files in the FORTRAN Usage Manual.

--FATAL FL 253-- Hollerith constant is missing terminating "".

Condition Identifier: FLE\$INCOMPLETE_HOLLERITH Code: FL 253

Description: A constant in list directed input must end with a quote (") symbol. The quote must be on the same line as the constant.

User Action: Check for a constant that is continued on a subsequent line or add a quote after the constant.

Further Information: Refer to the description of list directed input in the FORTRAN Usage Manual.

--FATAL FL 254-- Complex constant expected following '('.

Condition Identifier: FLE\$COMPLEX_CONST_INCOMPLETE Code: FL 254

Description: Either the real or the imaginary part of a complex constant is not correct.

User Action: Check for an illegal character in either the real or the imaginary part, a missing comma between the two parts, or a missing right parenthesis after the two parts.

Further Information: Refer to the description of complex constants in the FORTRAN Usage Manual.

--FATAL FL 255-- Comma expected in complex constant.

Condition Identifier: FLE\$COMPLEX_COMMA_MISSING Code: FL 255

Description: The first non-blank character after the real part of a complex constant must be a comma.

User Action: Insert a comma between the real and imaginary parts.

Further Information: Refer to the description of complex constants in the FORTRAN Usage Manual.

--FATAL FL 256-- ')' expected at end of complex constant.

Condition Identifier: FLE\$COMPLEX_RIGHT_PAREN_MISSING Code: FL 256

Description: The right parenthesis that terminates a complex constant must appear on the same line as the imaginary part.

User Action: Check for an incorrect complex constant.

Further Information: Refer to the description of complex constants in the FORTRAN Usage Manual.

--FATAL FL 257-- Record {text} does not exist on {file}.

Condition Identifier: FLE\$EOI_DIR_FMTD_IN Code: FL 257

Description: The specified record number exceeds the largest record number on the specified direct access file.

User Action: Change the record number to a correct value.

--FATAL FL 258-- Record length of {file} is {text}. For list_directed_output, length must be at least {text}.

Condition Identifier: FLE\$RECL_TOO_SMALL_FOR_LIST Code: FL 258

Description: The record length you specified on the OPEN statement for the list directed output file was too small. (The record length must be specified as a number of characters.)

User Action: Increase the record length of the specified file to a value greater than or equal to 45 characters.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 259-- A logical variable must contain a logical value.

Condition Identifier: FLE\$NOT_LOGICAL_VALUE Code: FL 259

Description: Your program attempted to read a non-logical value into a type logical variable.

User Action: Either specify a logical value for the input value or change the variable type to match the type of the input value. The logical value true is specified as T or .T followed by optional nonblank characters. The logical value false is specified as F or .F followed by optional nonblank characters.

Further Information: Refer to the description of namelist input in the FORTRAN Language Definition manual.

--FATAL FL 260-- Min number of digits must be \leq field width.

Condition Identifier: FLE\$M_GREATER_THAN_W Code: FL 260

Description: In the I, O, or Z descriptor, the minimum number of digits to be displayed, m, (indicated by the number to the right of the decimal point) must be less than or equal to the total width of the field, w, (indicated by the number to the left of the decimal point).

User Action: Change the edit descriptor so that m is less than or equal to w.

Further Information: Refer to the description of the I, O, or Z edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 261-- For scale factor {text}, number of fractional digits must be \geq {text}.

Condition Identifier: FLE\$D_TOO_SMALL Code: FL 261

Description: The scale factor, k, (indicated by the integer constant to the left of P) must be less than the number of digits to appear to the right of the decimal point, d, in the D, E, F, or G edit descriptor.

User Action: Increase the d field of the D, E, F, or G edit descriptor so that $d \geq |k-1|$.

Further Information: Refer to the description of the P edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 262-- Edit descriptor does not match data type {text}.

Condition Identifier: FLE\$EDIT_DESC_DATA_MISMATCH Code: FL 262

Description: The edit descriptor type in the format specification must match the item type in the formatted input or output list.

User Action: Check for an edit descriptor that does not match its associated list item. Common causes of this error are extra items in the input or output list, missing items in the input or output list, or extra edit descriptors in the format specification.

Further Information: Refer to the description of edit descriptors and of format specifications in the FORTRAN Usage Manual.

--FATAL FL 263-- Format nesting level must be $<$ {text}.

Condition Identifier: FLE\$NESTED_PAREN_LVL Code: FL 263

Description: The format specification has too many nested levels. That is, a specification of the form

```
FORMAT (...(...(...))...)
```

has too many sets of parentheses.

User Action: Rewrite the format specification to reduce the number of nested levels.

Further Information: Refer to the description of format specifications under Formatted Input/Output in the FORTRAN usage manual.

--FATAL FL 264-- No ending string delimiter has been seen.

Condition Identifier: FLE\$NO_ENDING_STR_DELIM Code: FL 264

Description: A string that must be enclosed in apostrophes (') or quote (") symbols is missing the terminating apostrophe or quote symbol.

User Action: Either add a terminating apostrophe or quote symbol, or check for a substring reference whose length is not large enough to include all the characters of the string up to the terminating apostrophe or quote symbol.

Further Information: Refer to the description of quote and apostrophe descriptors in the FORTRAN Usage Manual.

--FATAL FL 265-- Length of quoted string must be \leq {text}.

Condition Identifier: FLE\$QUOTED_STR_TOO_LARGE Code: FL 265

Description: The length of the string is larger than the maximum number of characters allowed in a character constant.

User Action: Check for a missing apostrophe or quote string delimiter.

Further Information: Refer to the description of character constants in the FORTRAN Usage Manual.

--FATAL FL 266-- + or - not followed by a number.

Condition Identifier: FLE\$ILLPLACED_SIGN Code: FL 266

Description: In a formatted input record, a + or - sign was found that was not followed by a number.

User Action: Check for an extra sign in the input record, or an incorrect number following the sign.

--FATAL FL 267-- Repeat spec must be < {text}.

Condition Identifier: FLE\$REPEAT_COUNT_TOO_LARGE Code: FL 267

Description: A repeat count generated more than the maximum number of characters allowed.

User Action: Either reduce the repeat count or write an equivalent format specification to specify a number of characters that is less than the maximum.

Further Information: Refer to the description of repeat counts under List Directed Input in the FORTRAN Usage Manual.

--FATAL FL 268-- Signed integer must be followed by P.

Condition Identifier: FLE\$SIGNED_INT_NOT_BEFORE_P Code: FL 268

Description: The P (scale factor) descriptor is the only descriptor that can be preceded by a signed integer.

User Action: Either insert a P after the signed integer or remove the sign from the integer.

Further Information: Refer to the description of the P edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 269-- P edit descriptor must be followed by ',', '/', ':', ')' or D, E, F, G edit descriptor.

Condition Identifier: FLE\$SCALE_FACTOR_POSITION Code: FL 269

Description: A P edit descriptor was found that was not followed by a separator or a floating-point edit descriptor.

User Action: Either supply the missing separator or correct the edit descriptor following the P edit descriptor.

Further Information: Refer to the description of the P edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 270-- W field of Aw edit descriptor must be <= {text}.

Condition Identifier: FLE\$A_WIDTH_TOO_LARGE Code: FL 270

Description: The value of the field width, w, in an A edit descriptor must not exceed the maximum number of characters allowed in a character string.

User Action: Either reduce the value of w, or write an equivalent format specification that does not exceed the maximum allowed number of characters.

Further Information: Refer to the description of the A edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 271-- B must be followed by N or Z.

Condition Identifier: FLE\$INVALID_B_EDIT_DESC Code: FL 271

Description: A B, which signals the beginning of a BN or BZ edit descriptor, must be followed by an N or Z.

User Action: Either correct the BN or BZ edit descriptor, or remove the B.

Further Information: Refer to the description of the BN or BZ edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 272-- H must be preceded by unsigned integer.

Condition Identifier: FLE\$H_NOT_PRECEDED_BY_# Code: FL 272

Description: An H, which signals the beginning of an H edit descriptor, must be preceded by an integer that specifies the length of the string, including blanks.

User Action: Supply a length specification in front of the H descriptor.

Further Information: Refer to the description of the H edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 273-- Value of n in nH edit descriptor > length of remaining format.

Condition Identifier: FLE\$FORMAT_OVERFLOW Code: FL 273

Description: The number that specifies the length of string is greater than the actual length of the string. This error could be caused either by a value of n that is too large, or by a string that is too short.

User Action: Either reduce the value of n or increase the length of the string.

Further Information: Refer to the description of the H edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 274-- P must be preceded by integer.

Condition Identifier: FLE\$P_NOT_PRECEDED_BY_# Code: FL 274

Description: A P (scale factor) descriptor must be preceded by an integer to specify the scale factor. The correct form is kP, where k is a signed or unsigned integer.

User Action: Supply a scale factor for the P descriptor.

Further Information: Refer to the description of the P descriptor under Formatted Input/Output in the FORTRAN Language Definition manual.

--FATAL FL 275-- Position value for tab edit descriptor must be < {text}.

Condition Identifier: FLE\$TAB_#_TOO_LARGE Code: FL 275

Description: The tab control value (the number after the T, TL, or TR) must be less than the maximum allowed value for a constant. A tab control value must be a nonzero unsigned decimal integer.

User Action: Decrease the tab control value.

Further Information: Refer to the description of the T, TL, TR edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 276-- T, TL, and TR must be followed by integer.

Condition Identifier: FLE\$INVALID_TAB_SPEC Code: FL 276

Description: A T, TL, or TR must be followed by a nonzero unsigned decimal integer that is less than 65,536. This integer represents a tab position.

User Action: Supply a tab control value after the T, TL, or TR.

Further Information: Refer to the description of the T, TL, TR edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 277-- X must be preceded by unsigned integer.

Condition Identifier: FLE\$X_NOT_PRECEDED_BY_# Code: FL 277

Description: An X, which signals an X edit descriptor, must be preceded by a nonzero unsigned integer that represents the number of character positions to be skipped from the current character position.

User Action: Precede the X by a nonzero unsigned integer.

Further Information: Refer to the description of the X edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 278-- Character in format not expected in this position.

Condition Identifier: FLE\$INVALID_EDIT_DESC Code: FL 278

Description: A character in the format specification is not recognizable as a character that begins an edit descriptor.

User Action: Check for extra characters in the format specification or a missing right parenthesis at the end of the format specification.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 279-- Repeat spec must not precede a nonrepeatable edit descriptor.

Condition Identifier: FLE\$RPT_COUNT_ON_NONRPT Code: FL 279

Description: A repeat count is not allowed on the BN, BZ, S, SS, SP, ', ", T, TL, TR, /, or : edit descriptors.

User Action: Either remove the repeat count or supply the correct edit descriptor.

Further Information: Refer to the description of the repeatable and nonrepeatable edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 280-- Separator must follow edit descriptor.

Condition Identifier: FLE\$SEPARATOR_MISSING Code: FL 280

Description: Edit descriptor specifications in a format specification must be separated by a comma, slash, or colon.

User Action: Supply a separator.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 281-- Fractional digits field of edit descriptor missing.

Condition Identifier: FLE\$DECIMAL_FIELD_MISSING Code: FL 281

Description: The number specifying the number of digits after the decimal point in a D, E, F, or G edit descriptor is missing. For example, in F6.2, 2 specifies that two digits follow the decimal point.

User Action: Supply the missing value.

Further Information: Refer to the description of the D, E, F, or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 282-- Fractional digits field of edit descriptor must be < {text}.

Condition Identifier: FLE\$d_FIELD_TOO_LARGE Code: FL 282

Description: The fractional digits field in a D, E, F, or G descriptor specifies the number of digits to appear after the decimal point in the output number. The value you specified for the d field is too large.

User Action: Supply a smaller value for the fractional digits field.

Further Information: Refer to the description of the D, E, F, or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 283-- Unsigned integer must follow E in edit descriptor.

Condition Identifier: FLE\$EXPONENT_WIDTH_MISSING Code: FL 283

Description: The value of the exponent e in an E (Ew.dEe) or G (Gw.dEe) edit descriptor is incorrect or missing.

User Action: Either specify an exponent or remove the E.

Further Information: Refer to the description of the E or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 284-- Exponent width must be < {text}.

Condition Identifier: FLE\$E_FIELD_TOO_LARGE Code: FL 284

Description: The e field (the number of digits in the exponent) in an Ew.dEe or Gw.dEe must be less than 6; however, only 4 digits are meaningful.

User Action: Reduce the nonzero unsigned integer constant following the E to a value less than 6.

Further Information: Refer to the description of edit descriptors in the FORTRAN Usage Manual.

--FATAL FL 285-- Exponent width field is valid only with E and G edit descriptors.

Condition Identifier: FLE\$E_FIELD_INVALID Code: FL 285

Description: Exponents must not be specified in a D or F edit descriptor.

User Action: Either insert a comma before the E to separate the D or F edit descriptor from the E edit descriptor, or remove the E.

Further Information: Refer to the description of either the D or the F edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 286-- W field must be < {text}.

Condition Identifier: FLE\$W_FIELD_TOO_LARGE Code: FL 286

Description: The field width, w, must be < 256 characters.

User Action: Either write an equivalent FORMAT using a tab control edit descriptor (X, T, TL, or TR), or check for a typing error in the w field.

Further Information: Refer to the description of the X, T, TL, or TR edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 287-- Absolute value of scale factor must be < {text}.

Condition Identifier: FLE\$SCALE_FACTOR_TOO_LARGE Code: FL 287

Description: The scale factor, k, is larger than the maximum allowed value for a constant.

User Action: Either check for a typing error in the scale factor (the integer constant before the P), or reduce the value of k.

Further Information: Refer to the description of the P edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 288-- W field of edit descriptor must be unsigned integer.

Condition Identifier: FLE\$W_FIELD_MISSING Code: FL 288

Description: The field width, w, must be a nonzero unsigned integer constant.

User Action: Check for a nondigit in the first nonblank position after the edit descriptor letter.

Further Information: Refer to the description of the D, F, G, I, O, or Z edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 289-- M field of edit descriptor must be unsigned integer.

Condition Identifier: FLE\$MISSING_M_FIELD Code: FL 289

Description: The number specifying the minimum number of digits to be output, m, must be an unsigned integer less than 256.

User Action: Check for an incorrectly placed decimal point in the edit descriptor, or remove the sign from the m field.

Further Information: Refer to the description of the I, O, or Z edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 290-- M field must be < {text}.

Condition Identifier: FLE\$M_FIELD_TOO_LARGE Code: FL 290

Description: The minimum number of digits to be output, m, must be an unsigned integer less than 256.

User Action: Check for an error in the m field of the edit descriptor.

Further Information: Refer to the description of the I, O, or Z edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 291-- Final ')' missing from format spec.

Condition Identifier: FLE\$FORMAT_TERMINATOR_MISSING Code: FL 291

Description: The end of the format specification was reached without finding a terminating right parenthesis.

User Action: Check for a missing right parenthesis at the end of the format specification.

Further Information: Refer to the description of format specifications in the FORTRAN Usage Manual.

--FATAL FL 292-- Insufficient space available for encoded format.

Condition Identifier: FLE\$NO_SPACE_FOR_FORMAT Code: FL 292

Description: An internal routine exceeded the bounds of a segment (2**31 bytes) while transforming a format specification into an internal representation.

User Action: Follow site-defined procedures for reporting software errors.

--FATAL FL 293-- Format contains empty format spec.

Condition Identifier: FLE\$INTERNAL_FORMAT_SPEC_EMPTY Code: FL 293

Description: An empty nested format specification (a pair of parentheses with nothing in between) was found.

User Action: Remove the empty format specification.

Further Information: Refer to the description of format specifications in the FORTRAN Usage Manual.

--FATAL FL 294-- Insufficient space available for decoded format.

Condition Identifier: FLE\$NO_SPACE_FOR_DECODE Code: FL 294

Description: An internal routine exceeded the bounds of a segment (2**31 bytes) while transforming an internal representation of a format specification into a character string.

User Action: Follow site-defined procedures for reporting software errors.

--FATAL FL 295-- Number of elements {text} exceeds max array size of {text}.

Condition Identifier: FLE\$ARRAY_TOO_BIG Code: FL 295

Description: The number of elements in an array has exceeded the maximum allowed number.

User Action: Check the bounds of an array specification or an adjustable dimension array and decrease the size of the array bounds. For example:

```
SUBROUTINE SUB (A, M, N)
```

```
    DIMENSION A(M,N)    where (N*M) > 16,777,216
```

Further Information: Refer to the description of adjustable dimensions in the FORTRAN Usage Manual.

--FATAL FL 296-- First and last word of DUMP must be in same segment.

Condition Identifier: FLE\$DUMP_ACROSS_SEGMENT Code: FL 296

Description: The arguments that specify the first and last words of the area to be dumped are contain an error.

User Action: Check the DUMP call for an incorrect array reference or an array reference that is out of bounds (exceeds the declared dimension).

Further Information: Refer to the description of the DUMP call in the FORTRAN Usage Manual.

--FATAL FL 297-- DUMP format indicator ({text}) must be in range 0..7.

Condition Identifier: FLE\$DUMP_FORMAT_BAD Code: FL 297

Description: The decimal integer that specifies the format of the dump is missing or incorrect. 0 indicates a hexadecimal dump, 1 indicates a real dump, 2 indicates an integer dump, and 3 indicates an octal dump.

User Action: Either supply the missing argument (it should be the third argument in the argument list) or correct the invalid argument.

Further Information: Refer to the description of DUMP in the FORTRAN Usage Manual.

--FATAL FL 298-- Integer value expected.

Condition Identifier: FLE\$INTEGER_EXPECTED Code: FL 298

Description: The I edit descriptor in formatted input requires a corresponding integer in the input record; a value that contains other than decimal digits, blanks, and a sign (optional) was found.

User Action: Check the I descriptor, the input list in the READ statement, and the input record. The I descriptor should correspond to an integer item in the input list and the input record, and the number of digits in the input number should match the field width w of Iw.

Further Information: Refer to the description of the I descriptor and of input/output lists in the FORTRAN Usage Manual.

--FATAL FL 299-- Signed number must contain a digit.

Condition Identifier: FLE\$NO_DIGIT_IN_SIGNED_NUM Code: FL 299

Description: The sign found in the formatted input record must be followed by a digit.

User Action: Either supply a digit after the sign or check the field width w of the edit descriptor; it must be large enough to include the sign and digits.

--FATAL FL 300-- An octal number is expected.

Condition Identifier: FLE\$OCTAL_NUMBER_EXPECTED Code: FL 300

Description: An octal constant is a string of 1 through 22 octal digits. An octal digit is one of the following digits: 0, 1, 2, 3, 4, 5, 6, or 7. If all 22 digits are used, the leftmost digit must be 0 or 1.

User Action: Retype the constant.

Further Information: Refer to the description of octal constants in the FORTRAN Usage Manual.

--FATAL FL 301-- A hexadecimal number is expected.

Condition Identifier: FLE\$HEX_NUMBER_EXPECTED Code: FL 301

Description: A hexadecimal constant is a string of 1 through 16 hexadecimal digits. A hexadecimal digit is one of the following characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F (lowercase letters are permissible).

User Action: Retype the constant.

Further Information: Refer to the description of hexadecimal constants in the FORTRAN Usage Manual.

--FATAL FL 302-- A logical value is expected.

Condition Identifier: FLE\$LOGICAL_VALUE_EXPECTED Code: FL 302

Description: A logical constant is one of the following: .TRUE., .FALSE., .T., and .F.

User Action: Correct the logical constant.

Further Information: Refer to the description of the L edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 303-- A floating point number is expected.

Condition Identifier: FLE\$FLOATING_NUMBER_EXPECTED Code: FL 303

Description: The D, E, F, or G edit descriptor must correspond to a decimal number in the input record. The probable cause of this error is an incorrect exponent in the decimal number. An exponent must be a nonzero unsigned integer constant with less than 7 digits, starting with a D, d, E, e, or sign (+ or -).

User Action: Check the number for an incorrect exponent.

Further Information: Refer to the description of the D, E, F, or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 304-- The mantissa must contain a digit.

Condition Identifier: FLE\$MANTISSA_REQUIRES_DIGITS Code: FL 304

Description: The mantissa (the number to the left of the exponent) in a decimal input value is incorrect or missing.

User Action: Supply a mantissa in front of the E that signals the beginning of the exponent.

Further Information: Refer to the description of the E or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 305-- The exponent must contain a digit.

Condition Identifier: FLE\$EXPONENT_REQUIRES_DIGITS Code: FL 305

Description: The exponent in an input value corresponding to a D, E, F, or G edit descriptor (the number after the D, E, or sign that signals the beginning of the exponent) is incorrect or missing.

User Action: Supply a nonzero unsigned integer constant with less than 7 digits as the exponent.

Further Information: Refer to the description of the D, E, F, or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 306-- First argument of WTSET must be a string of length 1.

Condition Identifier: FLE\$WTSET_1ST_ARG_LEN Code: FL 306

Description: The first argument in the WTSET call must be a character expression of length 1.

User Action: Supply a correct argument to WTSET.

Further Information: Refer to the description of the WTSET routine in the FORTRAN Usage Manual.

--FATAL FL 307-- Second argument of WTSET must be in range 0..255.

Condition Identifier: FLE\$WTSET_2ND_ARG_RANGE Code: FL 307

Description: The second argument in the WTSET routine call must be an integer expression with a value in the range 0..255.

User Action: Supply a correct argument to WTSET.

Further Information: Refer to the description of the WTSET routine in the FORTRAN Usage Manual.

--FATAL FL 308-- Internal file record length must be < {text}.

Condition Identifier: FLE\$INT_FILE_RECL_TOO_BIG Code: FL 308

Description: The length of an array used as an internal file exceeds the maximum allowed size. A possible cause is an adjustable dimension that yielded an excessively large value. For example:

```

SUBROUTINE SUB (A, M, N)

  DIMENSION A(N,M)   where (N*M) > 65,535.

```

This error could also be caused by the same problem with the dimension variables passed through COMMON.

User Action: Check the dimensions of the array.

Further Information: Refer to the description of internal files in the FORTRAN Usage Manual.

--FATAL FL 309-- Record length for NAMELIST output must be >= 72.

Condition Identifier: FLE\$RECL_TOO_SMALL_NAMEOUT Code: FL 309

Description: The record length of the namelist output file being written to must be at least 58 characters.

User Action: Be sure that you specified the correct file name. If correct, then change the record length of the file by using a SETFA command, OPEN statement, or PROGRAM statement.

Further Information: Refer to the description of the OPEN statement format, the SET_FILE_ATTRIBUTE command, or the PROGRAM statement in the FORTRAN Usage Manual.

--FATAL FL 310-- FILE= spec must appear in an OPEN with STATUS='OLD' or 'NEW'.

Condition Identifier: FLE\$OPEN_OLD_NEW_WITH_NO_FILE Code: FL 310

Description: If an OPEN statement specifies STATUS='OLD' or STATUS='NEW', then it must also specify FILE='filename'.

User Action: Supply a FILE= specifier, remove the STATUS= specifier from the OPEN, or change the STATUS='OLD' or 'NEW' specification to 'UNKNOWN'.

Further Information: Refer to the description of OPEN in the FORTRAN Usage Manual.

--FATAL FL 311-- Comma missing {text}.

Condition Identifier: FLE\$COMMA_MISSING Code: FL 311

Description: A comma separator is missing from the input record.

User Action: Supply a comma either at the end of the previous record or between the items.

Further Information: Refer to the description of namelist input in the FORTRAN Usage manual.

--FATAL FL 312-- Variable name not in group {text}.

Condition Identifier: FLE\$NAME_NOT_IN_NAMELIST Code: FL 312

Description: A variable name in the namelist input record is not in the namelist group referenced by the READ statement.

User Action: Be sure that all variables in the namelist input group are declared in the NAMELIST statement for that group.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 313-- ')' missing.

Condition Identifier: FLE\$RIGHT_PAREN_MISSING Code: FL 313

Description: In a namelist input group, a right parenthesis must terminate an array element before the end-of-line.

User Action: Supply a right parenthesis after the array element specification.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 314-- Constant must precede end of record.

Condition Identifier: FLE\$CONSTANT_MISSING Code: FL 314

Description: In a namelist input group, a value must be assigned to an item= specification before the end-of-line.

User Action: Check namelist input for a missing value after the last item= specification.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 315-- '=' must follow variable name.

Condition Identifier: FLE\$EQUAL_SIGN_MISSING Code: FL 315

Description: In a namelist input record, an equal sign must follow a variable name. For example:

A=6.2

User Action: Supply an equal sign between the variable name and its value in the namelist input record.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 316-- Subscript missing.

Condition Identifier: FLE\$SUBSCRIPT_MISSING_NAMIN Code: FL 316

Description: A subscript is missing from a namelist input item. The probable cause of this error is two consecutive commas or a left parenthesis-comma combination in a data item.

User Action: Correct the namelist input item.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 317-- Too few subscripts. {text} has {text} dimensions.

Condition Identifier: FLE\$INSUFFICIENT_SUBSCRIPTS Code: FL 317

Description: The number of subscripts in a namelist input item exceeds the number dimensions declared for the item.

User Action: Increase the number of subscripts in the namelist input item to match the number of declared dimensions.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 318-- Invalid subscript delimiter.

Condition Identifier: FLE\$INVALID_SUBSCRIPT_DELIMITER Code: FL 318

Description: A comma or right parenthesis must terminate a subscript in namelist input.

User Action: Correct the namelist input item.

--FATAL FL 319-- Too many subscripts. {text} has {text} dimensions.

Condition Identifier: FLE\$TOO_MANY_SUBSCRIPTS Code: FL 319

Description: The number of subscripts in a namelist input item exceeds the number of dimensions declared for the item.

User Action: Decrease the number of subscripts in the namelist input item to match the number of dimensions declared for the item.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 320-- Repeat count with no variable name.

Condition Identifier: FLE\$REPEAT_CNT_NO_VAR_NAME Code: FL 320

Description: In the namelist input record, a repeat count appeared without a constant or variable name.

User Action: Either remove the repeat count or specify a constant or variable name after the repeat count.

Further Information: Refer to the description of namelist input in the FORTRAN Language Definition manual.

--FATAL FL 321-- Hollerith constant must contain \leq 8 characters.

Condition Identifier: FLE\$HOLL_LONGER_THAN_8 Code: FL 321

Description: Under NOS/VE, a Hollerith constant must not exceed 8 characters. Under NOS, a Hollerith could contain up to 10 characters.

User Action: Check for a Hollerith constant in namelist input longer than 8 characters.

Further Information: Refer to the description of Hollerith constants in the FORTRAN Usage Manual.

--FATAL FL 322-- End of group. No data found.

Condition Identifier: FLE\$END_GROUP_NO_DATA Code: FL 322

Description: In a namelist input group, no data was found between the group name and the \$END terminator.

User Action: Check namelist input either for an extra dollar sign or ampersand that signals the end of namelist input, or for a missing record(s).

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 323-- Variable name and value not in same record.

Condition Identifier: FLE\$NAME_VALUE_NOT_SAME_LINE Code: FL 323

Description: In a namelist input group, an item and its assigned value must appear on the same line.

User Action: Check namelist input for an item that is continued on the next line.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 324-- Unrecognizable data.

Condition Identifier: FLE\$UNRECOGNIZABLE_DATA Code: FL 324

Description: In a namelist input record, an invalid value was assigned to an item.

User Action: Check input data for the following:

Misspelled logical constant

Missing equal assign

Missing left parenthesis after an equal sign

Illegal character in an input value

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 325-- Group name {text} not found before endfile on {file}.

Condition Identifier: FLE\$GROUP_NAME_NOT_FOUND Code: FL 325

Description: The group name specified in the namelist READ statement was not found in the file you specified. Possible errors include:

The group name does not exist on the file.

The group name is not preceded by a dollar sign or ampersand.

The group name does not appear in the proper position in the input record.

The group name occurs before the first record read by the READ statement.

The file was not positioned at BOI at the time of the READ.

User Action: Check for any of the preceding errors. Check for an incorrect or misspelled file name or group name. Be sure that the file is rewound before you read it. Be sure that the group name begins in column 2 of the input record.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 326-- Endfile on {file} encountered before end of group {text}.

Condition Identifier: FLE\$NAMELIST_ENDED_BY_EOF Code: FL 326

Description: The search for an item in namelist input was terminated by an end-of-file before a dollar sign or ampersand was found.

User Action: Insert a dollar sign or ampersand after the last comma in namelist input.

Further Information: Refer to the description of namelist input in the FORTRAN Usage Manual.

--FATAL FL 327-- Assigned GOTO variable must contain executable statement number.

Condition Identifier: FLE\$ASSIGNED_GOTO Code: FL 327

Description: The variable specified in an assigned GOTO statement did not contain the label of an executable statement. Possible causes:

The variable was not declared in a preceding ASSIGN statement.

The variable contains the label of a nonexecutable statement.

User Action: Specify the correct variable in the assigned GOTO statement, or specify an ASSIGN statement that assigns the label of an executable statement to the variable.

Further Information: Refer to the description of the ASSIGN statement in the FORTRAN Usage Manual.

--FATAL FL 328-- Starting character position must be > 0.

Condition Identifier: FLE\$SUBSTRING_LOWER_BOUND Code: FL 328

Description: The lower bound of the substring reference specifies a position before the first character position of the string.

User Action: Check for an uninitialized variable used as the lower bound of the substring reference, or for a substring reference that has a negative value.

Further Information: Refer to the description of substring references in the FORTRAN Usage Manual.

--FATAL FL 329-- JDATE must be declared CHARACTER*5.

Condition Identifier: FLE\$JDATE_STRLENGTH_ILLEGAL Code: FL 329

Description: The JDATE function must be declared CHARACTER*5.

User Action: Declare JDATE as CHARACTER*5 in your program.

Further Information: Refer to the description of the JDATE function in the FORTRAN Usage Manual.

--FATAL FL 330-- Value set number ({text}) of parameter {text} must be in range 1..{text}.

Condition Identifier: FLE\$SCL_SET_NUMBER_RANGE Code: FL 330

Description: In the SCL interface call, the argument that specifies the value set number was outside of the required range.

User Action: Checked for an uninitialized or mistyped argument.

Further Information: Refer to the description of SCL Interface Calls in the FORTRAN Usage Manual.

--FATAL FL 331-- Value number ({text}) of parameter {text} must be in range 1..{text}.

Condition Identifier: FLE\$SCL_VALUE_NUMBER_RANGE Code: FL 331

Description: In the SCL interface call, the argument that specifies the value number was outside the required range.

User Action: Check for an uninitialized or mistyped argument.

Further Information: Refer to the description of SCL Interface Calls in the FORTRAN Usage Manual.

--FATAL FL 332-- Keyword {text} unexpected for argument {text} of subroutine {text}.

Condition Identifier: FLE\$SCL_KEYWORD_BAD Code: FL 332

Description: The specified keyword value is not one of the values allowed for the specified argument to the SCL interface routine.

User Action: Check the spelling of the keyword.

Further Information: Refer to the description of SCL Interface Calls in the FORTRAN Usage Manual.

--FATAL FL 333-- {text} called to access {text} parameter.

Condition Identifier: FLE\$SCL_PARAM_KIND_MISMATCH Code: FL 333

Description: Each SCL parameter type (kind) requires different SCL interface call. The call you specified did not match the type of the parameter specified in the call.

User Action: Either change the SCL call to the correct call for the specified parameter, or change the SCL parameter definition to match the type required by the call.

Further Information: Refer to the description of SCL Interface Calls in the FORTRAN Usage Manual.

--FATAL FL 334-- {text} called to access {text} variable.

Condition Identifier: FLE\$SCL_VAR_KIND_MISMATCH Code: FL 334

Description: Each SCL variable type requires an variable interface call that matches that type. The call you specified did not match the SCL variable type being accessed.

User Action: Be sure that the SCL variable was created correctly using the CREV call. Then change either the variable type or the variable interface call to achieve a correct match.

Further Information: Refer to the description of the SCL Interface Calls in the FORTRAN Usage Manual.

--FATAL FL 335-- DATE must be declared CHARACTER*10.

Condition Identifier: FLE\$DATE_STRLENGTH_ILLEGAL Code: FL 335

Description: The DATE function must be declared CHARACTER*10.

User Action: Declare DATE as CHARACTER*10 in your program.

Further Information: Refer to the description of the DATE function in the FORTRAN Usage Manual.

--FATAL FL 336-- CLOCK and TIME must be declared CHARACTER*8.

Condition Identifier: FLE\$TIME_STRLENGTH_ILLEGAL Code: FL 336

Description: The CLOCK and TIME functions must be declared CHARACTER*8.

User Action: Declare CLOCK or TIME as CHARACTER*8 in your program.

Further Information: Refer to the description of the CLOCK or TIME function in the FORTRAN Usage Manual.

--FATAL FL 337-- String length must be > 0.

Condition Identifier: FLE\$SUBSTRING_LENGTH Code: FL 337

Description: The upper bound of a substring reference must be greater than or equal to the lower bound.

User Action: Check for a reversed or mistyped substring bounds specification, or for a substring bounds specification that has been overwritten.

--FATAL FL 338-- Subscript must be in range {text}..{text}.

Condition Identifier: FLE\$SUBSCRIPT_OUT_OF_RANGE Code: FL 338

Description: The value of a subscript expression must be within the bounds declared for that subscript in the DIMENSION, type, or COMMON statement.

User Action: Correct the subscript expression so that it is within the declared bounds.

--FATAL FL 581-- Unit {text} already open on call to OPENMS.

Condition Identifier: FLE\$MS_UNIT_OPENED Code: FL 581

Description: The unit (or file associated with the unit) specified in the OPENMS statement was already opened.

User Action: Either close the unit using or CLOSMS before issuing OPENMS, or specify a different unit.

Further Information: Refer to the description of OPENMS or CLOSMS in the FORTRAN Usage Manual.

--FATAL FL 582-- Name key must not be 0.

Condition Identifier: FLE\$MS_NAME_KEY_ZERO Code: FL 582

Description: The value of a name key in a READMS or WRITMS statement must not be zero.

User Action: Use another name key for the READMS or WRITMS.

Further Information: Refer to the description of READMS or WRITMS in the FORTRAN Usage Manual.

--FATAL FL 583-- Mass storage attempted on {text} file {file}.

Condition Identifier: FLE\$MS_ON_NON_MS_UNIT Code: FL 583

Description: A CLOSMS, READMS, WRITMS, or STINDX operation was attempted on a non-mass storage unit. Mass storage operations can be performed only on units opened for mass storage input/output by an OPENMS call.

User Action: Change the unit specified in the READMS, WRITMS, CLOSMS, or STINDX call to a mass storage unit.

Further Information: Refer to the description of mass storage input/output in the FORTRAN Usage Manual.

--FATAL FL 584-- Master index for {file} must be > = {text}.

Condition Identifier: FLE\$MS_INDEX_TOO_SMALL Code: FL 584

Description: The index array size specified in the OPENMS call for an existing file must be large enough to contain the number of entries in the file index (that is, it must be at least as large as the file index).

User Action: Increase the length of the index specified in the OPENMS call.

Further Information: Refer to the description of OPENMS in the FORTRAN Usage Manual.

--FATAL FL 585-- Mass storage file ({file}) must not be closed with CLOSE statement.

Condition Identifier: FLE\$CLOSE_STMT_WITH_MS_FILE Code: FL 585

Description: You attempted to close a mass storage file with a CLOSE statement. A mass storage file can be closed only with a CLOSMS call.

User Action: Replace the CLOSE statement by a CLOSMS call.

Further Information: Refer to the description of CLOSMS in the FORTRAN Usage Manual.

--FATAL FL 586-- SYSTEM utility called with FORTRAN error {text}: {text}

Condition Identifier: FLE\$SYSTEM_CALL Code: FL 586

Description: This error was caused by a call to runtime routine SYSTEM.

User Action: The appropriate action depends on the particular error (the text of which appears in the message you just received). If you are running an application, consult the author of the application.

--FATAL FL 587-- Number with decimal point must contain a digit.

Condition Identifier: FLE\$NO_DIGIT_IN_DOTTED_NUM Code: FL 587

Description: An input data field being processed by a D, E, F, or G edit descriptor contains a decimal point but no digits.

User Action: Check input data for errors. Possible errors include missing digits or fields on the record that are not in proper alignment with the edit descriptors.

Further Information: Refer to the description of the D, E, F, or G edit descriptor in the FORTRAN Usage Manual.

--FATAL FL 588-- Record size of {text} exceeded for formatted I/O on internal file.

Condition Identifier: FLE\$RECORD_SIZE_INTERNAL_FILE Code: FL 588

Description: The record length of an internal file is equal to the length (characters) of each element of the character array being used as the internal file. The formatted I/O operation attempted to store a value into the file that exceeded the record length.

User Action: Correct the format specification, or increase the record length.

Further Information: Refer to the description of standard internal files in the FORTRAN Usage Manual.

--FATAL FL 606-- Number of elements {text} exceeds max array size of {text}.

Condition Identifier: FLE\$ARRAY_TOO_LARGE Code: FL 606

Description: The probable cause of this error is an adjustable dimension in a subprogram that exceeded the maximum allowed array size. For example:

```
SUBROUTINE SUB (A, N, M)
```

```
    DIMENSION A(N,M)    where (N*M) > 16,777,216.
```

This error could also be caused by the same problem with the array dimensions passed through COMMON. The error is detected when the array is referenced.

User Action: Specify correct values for the array dimensions.

Further Information: Refer to the description of adjustable dimensions in the FORTRAN Usage Manual.

--WARNING FL 607-- Boolean input from C170 file {file}.

Condition Identifier: FLE\$BOOLEAN_INPUT_FROM_170 Code: FL 607

Description: Your NOS/VE program read a boolean value from a NOS or NOS/BE file. The value was transferred bit-by-bit with no conversion. Since many uses of boolean data are machine dependent, the results of the input operation may not be what you expect.

User Action: Check the results of the input operation to verify that they are what you expect.

Further Information: Refer to one of the following manuals for information about migrating boolean data:

Migration From NOS to NOS/VE

Migration From NOS/BE to NOS/VE

--FATAL FL 608-- Record length of {text} must be <= internal file length of {text}.

Condition Identifier: FLE\$RECORD_LEN_GT_INTERNAL_FILE Code: FL 608

Description: The record length you specified in the internal file I/O operation exceeded the maximum allowed for the file.

User Action: Change the record length to a value less than the length defined for internal file.

Further Information: Refer to the description of internal files in the FORTRAN Language Definition manual.

--FATAL FL 610-- Mass storage I/O is not supported for C170 file {file}.

Condition Identifier: FLE\$MSIO_UNSUPPORTED_FOR_C170 Code: FL 610

Description: Your program uses the READMS statement to access the CYBER 170 file. You can use FMA only with the following kinds of I/O: formatted and unformatted reads (sequential and direct access), and list directed and NAMELIST reads (sequential).

User Action: Reconsider your migration plan for the application. The program must be changed to be used on NOS/VE.

Further Information: See the Migration Guide From NOS to NOS/VE manual for the file requirements for using FORTRAN FMA (in the discussion of FMA) and for information about changing mass storage I/O for use on NOS/VE (in the discussion of Migrating FORTRAN Programs).

--FATAL FL 611-- BUFFER I/O is not supported for C170 file {file}.

Condition Identifier: FLE\$BUFIO_UNSUPPORTED_FOR_C170 Code: FL 611

Description: Your program uses the BUFFER IN statement to access the file. FMA can read and migrate files using the following kind of input/output: formatted and unformatted reads (sequential and direct access), and list directed and NAMELIST reads (sequential).

User Action: Reconsider your migration plan for the application. The program must be changed to be used on NOS/VE.

Further Information: See the Migration From NOS to NOS/VE manual for the file requirements for using FORTRAN FMA (in the discussion of FORTRAN FMA) and for information about changing buffer I/O for use on NOS/VE (in the discussion of Migrating FORTRAN Programs).

--WARNING FL 612-- Boolean output to C170 file {file}.

Condition Identifier: FLE\$BOOLEAN_OUTPUT_TO_170 Code: FL 612

Description: Your NOS/VE program wrote a boolean value to a NOS or NOS/BE file. The value was transferred bit-by-bit with no conversion. Because most uses of boolean values are machine dependent, the results of the output operation may not be what you expect.

User Action: Check the results of the output operation to verify that they are what you expect.

Further Information: Refer to one of the following manuals for information on how FMA handles boolean data:

- Migration From NOS to NOS/VE
- Migration From NOS/BE to NOS/VE

--WARNING FL 613-- FMA detected data conversion error processing file {file}.

Condition Identifier: FLE\$FMA_CONVERSION_ERROR Code: FL 613

Description: A conversion error occurred while FMA was migrating a file from NOS or NOS/BE to NOS/VE.

User Action: This is a possible internal error. First, be sure that the input file contains no errors and that the I/O operation is correct for the file type. If the error persists, follow site-defined procedures for reporting internal errors.

Further Information: Refer to the description of the File Migration Aid in one of the following manuals:

- Migration From NOS to NOS/VE
- Migration From NOS/BE to NOS/VE

--FATAL FL 625-- Array {text} is already allocated.

Condition Identifier: FLE\$DUPLICATE_ALLOCATE Code: FL 625

Description: The indicated array was specified in more than one ALLOCATE statement without intervening DEALLOCATE statements. After an array is allocated, it cannot appear in another ALLOCATE statement unless it is first deallocated.

User Action: Check the placement and format of the ALLOCATE and DEALLOCATE statements in your program. An array is considered to be allocated after execution of the ALLOCATE statement referencing the array name. The array becomes deallocated when the same array name appears in a DEALLOCATE statement.

Check for correct spelling of the array name in the ALLOCATE and DEALLOCATE statements. If an array-valued external function is used in your program, check the function name in the ALLOCATE statement within the function.

Further Information: Refer to the description of allocatable arrays in the array chapter of the FORTRAN Version 2 usage manual.

--FATAL FL 626-- Array {text} must be allocated before being deallocated.

Condition Identifier: FLE\$UNALLOCATED_ARRAY Code: FL 626

Description: The array name specified in the DEALLOCATE statement was not allocated. An array is allocated when the array name appears in an ALLOCATE statement.

User Action: Check the array name specified in the DEALLOCATE statement. It should match the name specified in a previous ALLOCATE statement.

Further Information: Refer to the description of allocatable arrays in the FORTRAN Version 2 usage manual.

--FATAL FL 627-- Result array for array-valued function {text} must be allocated in the function.

Condition Identifier: FLE\$FUNC_RESULT_UNALLOCATED Code: FL 627

Description: The array-valued function name must be allocated in the function.

User Action: Check that the function name appears in an ALLOCATE statement within the function.

Further Information: Refer to the description of external functions in the FORTRAN Version 2 usage manual.

--FATAL FL 628-- Insufficient space to ALLOCATE {text} words.

Condition Identifier: FLE\$ALLOC_TOO_BIG Code: FL 628

Description: The array you specified in the ALLOCATE statement cannot be allocated.

User Action: Allocate the array at a smaller size if possible. A noncharacter array must not exceed (2**28)-1 (which equals 268,435,455) words in length. A character array must not exceed (2**31)-1 (which equals 2,147,483,647) bytes.

--FATAL FL 631-- OPEN attempted on already opened common block (unit) {text}.

Condition Identifier: FLE\$OPEN_ON_OPENED_COM_BLOCK Code: FL 631

Description: A segment access file that was previously opened was specified in an OPEN statement.

User Action: Check for an OPEN statement that opens the segment access file.

Further Information: Refer to the description of the OPEN statement in the Input/Output chapter in the FORTRAN Usage Manual.

--FATAL FL 632-- CLOSE attempted on unopened common block (unit) {text}.

Condition Identifier: FLE\$CLOSE_ON_UNOPENED_COM_BLOCK Code: FL 632

Description: A segment access file that was unopened was specified in a CLOSE statement.

User Action: Check the OPEN and CLOSE statements for the segment access file.

Further Information: Refer to the description of either OPEN or CLOSE in the FORTRAN Usage Manual.

--FATAL FL 633-- This text is replaced by the relevant common block file system error.

Condition Identifier: FLE\$COMMON_BLOCK_FILE_SYST_ERR Code: FL 633

Description: An abnormal status condition was returned from a system routine while processing a common block segment access file. The relevant error number and message from the system routine are displayed.

User Action: Consult the Diagnostics Messages for NOS/VE manual and follow the User Action described for the error specified. Also, check that the segment access file has been set up correctly in your program and that specifiers for the OPEN and CLOSE are correct.

Further Information: Refer to the appropriate usage manual as indicated by the product identifier associated with the error or inform the site analyst, providing appropriate documentation of the error.

--FATAL FL 634-- SIZE on common block segment file CLOSE must not be negative. Value is {text}.

Condition Identifier: FLE\$CLOSE_SIZE_NOT_POSITIVE Code: FL 634

Description: The SIZE specifier on a CLOSE statement for a common block segment file must not be negative.

User Action: Correct the value of the SIZE specifier.

Further Information: Refer to the description of the CLOSE statement in the FORTRAN Usage Manual.

--FATAL FL 635-- SIZE on common block segment file CLOSE must be < {text}.

Condition Identifier: FLE\$CLOSE_SIZE_TOO_LARGE Code: FL 635

Description: The SIZE specifier on a CLOSE statement for a segment access file is too large.

User Action: Correct the SIZE specifier.

Further Information: Refer to the description of CLOSE in the FORTRAN Usage Manual.

--FATAL FL 636-- At line {text} of {text}, the lower bound provided for subscript {text} of {text} is {text}. This must be at least as big as the declared lower bound ({text}).

Condition Identifier: FLE\$SECT_BEGIN_TOO_SMALL Code: FL 636

Description: The lower bound of the array section is less than the lower bound of the corresponding dimension of the array being sectioned. The section selector must not reference an element outside the bounds of the array being sectioned.

User Action: Correct the format of the array section reference; check that you are referencing the correct dimension of the array.

Further Information: Refer to the array section description in the array chapter of the FORTRAN Version 2 for NOS/VE usage manual.

--FATAL FL 637-- At line {text} of {text}, the upper bound provided for subscript {text} of {text} is {text}. This must be no larger than the declared upper bound ({text}).

Condition Identifier: FLE\$SECT_END_TOO_BIG Code: FL 637

Description: The upper bound of a section selector is greater than the upper bound of the corresponding dimension of the array being sectioned. The section selector must not reference an element outside the bounds of the array being sectioned.

User Action: Correct the format of the array section reference; check that you are referencing the correct dimension of the array.

Further Information: Refer to the array section description in the array chapter of the FORTRAN Version 2 for NOS/VE usage manual.

--FATAL FL 638-- At line {text} of {text}, an element of the vector section selector for dimension {text} of {text} has value {text}. This must be within the declared bounds of {text} and {text}.

Condition Identifier: FLE\$VEC_SECT_SELECTOR_BOUNDS Code: FL 638

Description: The array you specified as a section selector has caused a reference outside the bounds of the array being sectioned. The elements in the section selector s of the array must lie within the bounds of the corresponding dimension of the array being sectioned.

User Action: Check the elements in the section selector array. Also, check the dimension bound declarations for the array being sectioned. For example, given the following statements:

```
integer a(1:10), b(2), c(2)
data b/11, -2/
data c/3,7/
```

The array section reference a(c) is valid because a(3) and a(7) are within the range of a. However, the array section reference a(b) is not valid because a(11) and a(-2) are out of the range of array a.

Further Information: Refer to the description of array sections in the FORTRAN Version 2 usage manual.

--FATAL FL 639-- At line {text} of {text}, there is an array operator whose operands do not conform. The extent of dimension {text} of {text} is {text}, while the extent of {text} is {text}. The extents must be the same.

Condition Identifier: FLE\$OPERANDS_DO_NOT_CONFORM Code: FL 639

Description: The operands in the array expression do not conform. Two array objects are conformable if they have the same number of dimensions and if corresponding dimensions are of the same size.

User Action: If you have two array operands, make sure they are conformable by checking the number of dimensions and the size of each dimension. Also, the operands must both be of type logical or both be of arithmetic type.

Further Information: Refer to the description of array conformability in the array chapter of the FORTRAN Version 2 for NOS/VE usage manual.

--FATAL FL 640-- At line {text} of {text}, dimension {text} of {text} has extent {text}, but {text} has extent {text}. The extents must be the same.

Condition Identifier: FLE\$ASSIGN_OPS_DO_NOT_CONFORM Code: FL 640

Description: The array objects in the assignment statement are not conformable. In an array-valued assignment statement, all array objects must have the same number of dimensions, and the corresponding dimensions must have the same size.

User Action: Check the number of dimensions and the size of each dimension for all array objects in the assignment statement. Also check array declarations.

Further Information: Refer to the description of assignment statements in the FORTRAN Version 2 for NOS/VE usage manual.

--FATAL FL 641-- {text} has at least one assumed-shape dummy argument. The calling routine must be compiled by VECTOR_FORTRAN.

Condition Identifier: FLE\$ASSMD_SHAPE_ARG_ERR Code: FL 641

Description: Your program contains an assumed-shape array as a dummy argument.

User Action: Since only FORTRAN Version 2 accepts assumed-shape arrays, you must recompile your program using the VECTOR_FORTRAN compilation command so that the assumed-shape dummy argument will be recognized.

As an alternative, you can assign upper bounds to the dummy array argument in the subprogram and compile with the FORTRAN compilation command.

Further Information: Refer to the description of declaring arrays in chapter 3 of the FORTRAN Version 2 for NOS/VE manual. The VECTOR_FORTRAN compilation command is described in chapter 11.

--FATAL FL 642-- Insufficient space remains to process another input/output statement.

Condition Identifier: FLE\$CILIST_SPACE_OVERFLOW Code: FL 642

Description: Probable memory allocation error.

User Action: Follow site-defined procedures for reporting software errors or operational problems.

--FATAL FL 644-- FMA does not support {text}-byte INTEGER values.

Condition Identifier: FLE\$FMA_WITH_INT_STAR_N Code: FL 644

Description: The File Migration Aid does not support integer*length data.

User Action: Change the integer values to full-word (8 byte) integers.

Further Information: Refer to the description of the INTEGER statement in the FORTRAN Language Definition manual.

--FATAL FL 645-- {text}-byte INTEGER values must be in the range from {text} through {text}.

Condition Identifier: FLE\$INT_STAR_N_RANGE Code: FL 645

Description: The size of the integer is not large enough for the value being assigned.

User Action: Change the size of the integer. The following table indicates the allowable ranges for integer (and type byte) sizes:

Integer Size	Range
8 byte	$-(2^{**63})$ through $(2^{**63})-1$
4 byte	$-(2^{**31})$ through $(2^{**31})-1$
2 byte	$-(2^{**15})$ through $(2^{**15})-1$
1 byte (byte data)	$-(2^{**7})$ through $(2^{**7})-1$

Further Information: Refer to the description of integer constants, integer variables, and the INTEGER statement in the FORTRAN Language Definition manual.

--FATAL FL 646-- CONDNAM must be declared CHARACTER*31.

Condition Identifier: FLE\$CONDNAM_STRLLENGTH_ILLEGAL Code: FL 646

Description: The CONDNAM function must be declared as type character*31 in the calling program.

User Action: CONDNAM can be correctly declared by the following statement:

CHARACTER CONDNAM*31

--FATAL FL 647-- FMA does not support {text}-byte LOGICAL values.

Condition Identifier: FLE\$FMA_WITH_LOGICAL_STAR_N Code: FL 647

Description: The File Migration Aid does not support 1-, 2-, or 4-byte logical values.

User Action: Change the value to an 8-byte logical value.

Further Information: See the description of the LOGICAL statement in the FORTRAN Version 2 Usage manual.

--FATAL FL 648-- The variable for the NAME= spec is too short to receive the file reference requested with INQUIRE.

Condition Identifier: FLE\$NAME_VAR_TOO_SHORT_FOR_FILE Code: FL 648

Description: The character variable or array element is not large enough to contain the full path name for the file reference or its internally assigned file name.

User Action: Increase the length of the character variable or array element that is receiving the NAME file specification. The length of the character variable or the total length of the character array element should be at least 31 characters to guarantee that at least the internally assigned file name is returned. A length of 512 characters guarantees that the complete file reference is returned; otherwise, an internally assigned file name is returned if the file reference is too large for the associated character variable or array element.

Further Information: The NAME specifier on the INQUIRE statement is described in the Input/Output chapters of the FORTRAN Version 1 and Version 2 Usage manuals.

--ERROR FS 1-- {text} cannot be opened - only files can be opened.

Condition Identifier: FSE\$CANNOT_OPEN_CATALOGS Code: FS 1

Description: The file path you are attempting to open does not have enough elements to be a file. It appears you are attempting to open a catalog.

User Action: Specify a file name.

--ERROR FS 3-- File {file} : An outstanding instance of open authorizes access for {text} which conflicts with the specified access modes of {text}.

Condition Identifier: FSE\$CONCURRENT_ACCESS_CONFLICT Code: FS 3

Description: You are attempting to open a file that is already open in the job. The task which opened the file is not willing to share for the modes of access that you specified or that were required by the command you executed.

User Action: Close the outstanding instance of open. You may also need to detach the file.

--ERROR FS 5-- File {file} : The specified open share modes of {text} do not include the access_modes of {text} specified by outstanding instances of open.

Condition Identifier: FSE\$CONCURRENT_SHARE_CONFLICT Code: FS 5

Description: You are attempting to open a file that is already open within the job. The command you executed is not willing to share the file for the modes of access which have been granted the task that already has the file open.

User Action: Specify open share modes that include all the access modes of outstanding instances of open, or execute the command again when the outstanding instances of open have been closed.

--ERROR FS 6-- File {file} is already attached to this job for access modes and required share modes which conflict with the specified modes of access or share.

Condition Identifier: FSE\$REDUNDANT_ATTACH_CONFLICT Code: FS 6

Description: You are attempting to open a file that is already attached to the current job. The access and/or share modes that you specified or that were required by the command you executed, conflict with the access and/or share modes for which the file is already attached.

User Action: Make sure the access modes you specify match the attached file's access modes and specify share modes that include the current share modes of the file, or execute the command again when the file has been detached.

--ERROR FS 8-- File {file} : Specifying private_read as true is improper for this instance of open because the access_mode is not READ or (READ, EXECUTE).

Condition Identifier: FSE\$IMPROPER_PRIVATE_READ Code: FS 8

Description: You are attempting to open a file with access modes that include some form of WRITE, while specifying the value TRUE for PRIVATE_READ either on the FSP\$OPEN_FILE request or a preceding ATTACH_FILE command.

User Action: Specify access modes of READ or (READ, EXECUTE) or specify FALSE for PRIVATE_READ.

--ERROR FS 10-- File {file} : {text} and {text} cannot both be specified for the same instance of open.

Condition Identifier: FSE\$REDUNDANT_ATTACHMENT_CHOICE Code: FS 10

Description: On an instance of open, error exit procedure and error exit procedure name cannot both be specified.

User Action: Remove either error exit procedure or error exit procedure name from the list of attachment options specified.

--ERROR FS 15-- File {file} : The specification of {text} is inconsistent with a previous specification of the ATTACH_FILE command.

Condition Identifier: FSE\$REDUNDANT_ATTACHMENT_SPEC Code: FS 15

Description: The parameter specified by the error message had been specified on a previous attachment of the file and cannot be specified again until the file has been detached.

User Action: Either do not specify the parameter or detach the file.

--ERROR FS 20-- An unrecoverable error internal to the fs system has occurred. {text} {text} {text} {text} {text} {text} {text} {text} {text} {text}

Condition Identifier: FSE\$SYSTEM_ERROR Code: FS 20

Description: A system error internal to the file system has occurred. This is not a user error.

User Action: Write a PSR against the file system specifying the error message text and as much information as possible regarding the circumstances surrounding the error.

--ERROR FS 21-- The maximum number of temporary files allowed has been reached. Some temporary files should be deleted before continuing.

Condition Identifier: FSE\$TEMP_FILES_LIMIT_REACHED Code: FS 21

Description: The user has reached the maximum number of temporary files allowed in a job.

User Action: Delete some of the temporary files or logout.

--ERROR FS 31-- A cycle reference was specified for catalog "{file}", but catalogs do not have cycles.

Condition Identifier: FSE\$CATALOGS_DO_NOT_HAVE_CYCLES Code: FS 31

Description: You appended a cycle reference to a catalog path.

User Action: Remove the cycle reference or specify a file with the cycle reference.

--ERROR FS 34-- \$LOCAL cannot have subcatalogs.

Condition Identifier: FSE\$LOCAL_SUBCATALOG_ILLEGAL Code: FS 34

Description: The path you specified on the command or call had \$LOCAL as the master catalog and more than one name following \$LOCAL. The \$LOCAL catalog cannot have subcatalogs.

User Action: Specify only one name for \$LOCAL files.

--ERROR FS 40-- File {file} cannot be closed because it is the target of a file connection.

Condition Identifier: FSE\$CLOSE_OF_TARGET_NOT_ALLOWED Code: FS 40

Description: The file_identifier specified on the close was that of a target of a file connection. A target of a file connection cannot be closed directly.

User Action: Not applicable.

--ERROR FS 50-- FSP\$COPY_FILE internal system error has occurred: {text}.

Condition Identifier: FSE\$COPY_INTERNAL_ERROR Code: FS 50

Description: A system error internal to fsp\$copy_file has occurred. This is not a user error.

User Action: Write a PSR against fsp\$copy_file, specifying the error message text and as much information as possible regarding the two files specified on the copy, such as file attributes, file connections, and so on.

--ERROR FS 54-- The input file, {file}, is positioned at end of information.

Condition Identifier: FSE\$INPUT_FILE_AT_EOI Code: FS 54

Description: The open_position for the file to be copied must not be its end of information.

User Action: Ensure that the open_position is not end of information.

--ERROR FS 58-- File contents and/or file structure of the input file, {file}, conflicts with that of the output file, {file}.

Condition Identifier: FSE\$CONFLICTING_FILE_CONTENTS Code: FS 58

Description: To copy a file, the file contents and the file structure of the input file must be the same as the file contents and the file structure of the output file, or the file contents and file structure of one of the two files must be UNKNOWN. If one of the two files has a file contents of LEGIBLE, the other may have a file contents of LIST; in either case the file structure must be UNKNOWN or DATA.

User Action: Change the file contents and/or file structure of one of the files so that the preceding criteria are met.

--ERROR FS 60-- Statement identifier {text} of the input file, {file}, and the output file, {file}, are unequal.

Condition Identifier: FSE\$STATEMENT_IDENTS_UNEQUAL Code: FS 60

Description: If the file_contents attribute of both files is legible, the statement_identifier attribute values for both files must also match.

User Action: Determine the statement_identifier attribute value of the input file and change the statement_identifier attribute value of the output file to match that of the input file.

--ERROR FS 62-- Line number {text} of the input file, {file}, and the output file, {file}, are unequal.

Condition Identifier: FSE\$LINE_NUMBERS_UNEQUAL Code: FS 62

Description: If the file_contents attribute of both files is legible, the line_number attribute values for both files must also match.

User Action: Determine the line_number attribute value of the input file and change the line_number attribute value of the output file to match that of the input file.

--WARNING FS 64-- One or more records of the output file, {file}, were truncated.

Condition Identifier: FSE\$OUTPUT_RECORD_TRUNCATED Code: FS 64

Description: At least one record of the input file was truncated upon copying to the output file.

User Action: If the output file has a record type of fixed, specify a larger maximum record length on the output file. If the output file has a record type of undefined and a block type of user specified, specify a larger maximum block length on the output file.

--ERROR FS 66-- The input file, {file}, is empty.

Condition Identifier: FSE\$EMPTY_INPUT_FILE Code: FS 66

Description: The copy was not performed because the input file contains no data.

User Action: Check that the correct input file is specified.

--ERROR FS 70-- {file} has insufficient maximum record/block length to perform requested copy.

Condition Identifier: FSE\$INSUFFICIENT_RECORD_LENGTH Code: FS 70

Description: If a list to legible or legible to list conversion is to take place, the list file must have a maximum record/block length equal to or greater than 2, otherwise all other files must have a maximum record/block length equal to or greater than 1.

User Action: Increase the maximum record/block length of the specified file.

--ERROR FS 72-- Byte addressable files {file} and {file} are not positioned at the same byte address.

Condition Identifier: FSE\$CONFLICTING_FILE_ADDRESSES Code: FS 72

Description: To copy a file with byte addressable file organization, the current byte address of the output file must be the same as that of the input file.

User Action: Determine the current byte address of the input file and set the address of the output file so that it matches that of the input file. (An AMP\$SEEK_DIRECT call can set the byte address of the file.)

--ERROR FS 74-- If {file} has byte addressable file organization, then so must file {file}.

Condition Identifier: FSE\$IMPROPER_FO_FOR_COPY Code: FS 74

Description: When copying a file, if either the input or output file has a byte addressable file_organization, the other file must also have a byte addressable file_organization.

User Action: Set the file_organization attribute of the output file to byte addressable.

--ERROR FS 76-- Byte addressable file {file} does not reside on mass storage.

Condition Identifier: FSE\$COPY_DEVICE_CONFLICT Code: FS 76

Description: To copy a byte addressable file, both the input and output files must be assigned to the mass storage device class.

User Action: Change the device class assignment of the output file.

--ERROR FS 78-- Byte addressable files {file} and {file} do not have the same block_type.

Condition Identifier: FSE\$CONFLICTING_BLOCK_TYPES Code: FS 78

Description: To copy a file with byte addressable file organization, the block_type attribute of the output file must be the same as that of the input file.

User Action: Set the block_type attribute of the output file to that of the input file.

--ERROR FS 80-- Byte addressable files {file} and {file} do not have the same record type.

Condition Identifier: FSE\$CONFLICTING_RECORD_TYPES Code: FS 80

Description: To copy a file with byte addressable file organization, the record_type attribute of the output file must be the same as that of the input file.

User Action: Determine the record_type attribute value of the input file and change the record_type attribute of the output file so that it matches that of the input file.

--ERROR FS 82-- Byte addressable files {file} and {file} do not have the same file access procedure name.

Condition Identifier: FSE\$FAP_NAMES_NOT_IDENTICAL Code: FS 82

Description: To copy a file with byte addressable file organization, the file access procedure name of the input file must be the same as that of the output file.

User Action: Change the file access procedure name of one of the files so that both are the same.

--ERROR FS 84-- Fixed length byte addressable files {file} and {file} do not have the same maximum record length.

Condition Identifier: FSE\$RECORD_LENGTHS_UNEQUAL Code: FS 84

Description: To copy a file with byte addressable file organization and record type of fixed, the maximum record length of the output file must be the same as that of the input file.

User Action: Set the maximum record length of the output file to that of the input file.

--ERROR FS 86-- Fixed length byte addressable files {file} and {file} do not have the same maximum block length.

Condition Identifier: FSE\$BLOCK_LENGTHS_UNEQUAL Code: FS 86

Description: To copy a file with byte addressable file organization, record type of undefined, and a block type of user specified, the maximum block length of the output file must be the same as that of the input file.

User Action: Set the maximum block length of the output file to that of the input file.

--ERROR FS 87-- A byte addressable file cannot be copied if the size of the input file, {file}, is larger than the size limit of the output file, {file}.

Condition Identifier: FSE\$SIZE_EXCEEDS_OUTPUT_LIMITS Code: FS 87

Description: To copy a file with byte addressable file organization, the file limit attribute of the output file must be equal to or larger than the size of the input file.

User Action: Change the file limit of the output file so that it is larger than (or equal to) the size of the input file.

--ERROR FS 88-- Fixed length byte addressable files {file} and {file} do not have the same padding character.

Condition Identifier: FSE\$PADDING_CHARACTERS_UNEQUAL Code: FS 88

Description: To copy a file with byte addressable file organization and a record type of fixed, both files must have the same padding character.

User Action: Set the padding character on the output file so that it is the same as the input file.

--ERROR FS 89-- Trailing character delimited/byte addressable files {file} and {file} do not have the same trailing character.

Condition Identifier: FSE\$TRAILING_CHARACTERS_UNEQUAL Code: FS 89

Description: To copy a file with byte address and record type of TRAILING_CHARACTER_DELIMITED, both files must have the same trailing character.

User Action: Set the trailing character on the output file so it is the same as the input file.

--ERROR FS 90-- If the input file, {file}, has undefined record type and system specified blocking, then it cannot have file contents of list.

Condition Identifier: FSE\$FROM_LIST_INPUT_UNSUPPORTED Code: FS 90

Description: This combination of record type, block type, and file contents is not allowed on a copy.

User Action: Change the file contents of the input file to unknown or to the file contents of the output file.

--ERROR FS 92-- If the output file, {file}, has undefined record type and system specified blocking, then it cannot have file contents of list.

Condition Identifier: FSE\$TO_LIST_OUTPUT_UNSUPPORTED Code: FS 92

Description: This combination of record type, block type, and file contents is not allowed on a copy.

User Action: Change the file contents of the output file to unknown or to the file contents of the input file.

--ERROR FS 94-- If the input file, {file}, has undefined record type and system specified blocking, then the output file, {file}, cannot have file contents of list.

Condition Identifier: FSE\$TO_LIST_INPUT_UNSUPPORTED Code: FS 94

Description: Conversion to a list file is not allowed from a file with undefined record type and system specified blocking.

User Action: Change the file contents of the output file to unknown or to the file contents of the input file.

--WARNING FS 96-- The FILE_STRUCTURE of the output file, {file}, has been truncated as a result of FSP\$COPY_FILE. ORIGINAL FILE_CONTENTS = "{text}". ORIGINAL FILE_STRUCTURE = "{text}".

Condition Identifier: FSE\$OUTPUT_STRUCTURE_TRUNCATED Code: FS 96

Description: When the combined lengths of FILE_CONTENTS and FILE_STRUCTURE is greater than 30 characters, the FILE_STRUCTURE is truncated when it is merged with FILE_CONTENTS into the FILE_CONTENTS field of the FILE_CONTENT_AND_PROCESSOR attribute for FSP\$OPEN_FILE.

User Action: Choose a FILE_CONTENTS and FILE_STRUCTURE that have a combined length of 30 characters or less.

--ERROR FS 100-- File information retrieval - internal system error has occurred: {text}.

Condition Identifier: FSE\$GET_FILE_INFO_INTERNAL Code: FS 100

Description: An internal system error has occurred. This is not a user error.

User Action: Write a PSR against the interface/command that caused the error, specifying the error message text and as much information as possible regarding the file specified on the interface/command (such as file attributes, file connections, and so on).

--WARNING FS 102-- The FILE_CONTENTS of file, {file}, has been truncated as a result of merging of FILE_CONTENTS and FILE_STRUCTURE in {text}. ORIGINAL FILE_CONTENTS = "{text}". ORIGINAL FILE_STRUCTURE = "{text}".

Condition Identifier: FSE\$FILE_CONTENTS_TRUNCATED Code: FS 102

Description: When the combined lengths of FILE_CONTENTS and FILE_STRUCTURE is greater than 30 characters, the FILE_STRUCTURE is truncated when it is merged with FILE_CONTENTS into the FILE_CONTENTS field of the FILE_CONTENT_AND_PROCESSOR attribute for FSE\$OPEN_FILE.

User Action: Choose a FILE_CONTENTS and FILE_STRUCTURE that have a combined length of 30 characters or less.

--ERROR FS 105-- The {text} sequence passed to {text} for the file, {file}, is insufficient in size for the requested operation.

Condition Identifier: FSE\$INSUFFICIENT_SEQUENCE_SIZE Code: FS 105

Description: The size of the user_defined_attributes sequence specified on the file information retrieval interface is less than the size of user defined attributes of the file.

User Action: Repeat the call using a sequence that is at least as large as the size returned by the user_defined_attribute_size parameter.

**--WARNING HU 5-- define_prior command must be supplied for the first screen.
Prior set to point to {text}.**

Condition Identifier: HUE\$FIRST_PRIOR_MUST_EXIST Code: HU 5

Description: The first screen in the source file must contain a DEFINE_PRIOR directive because of the way the CREATE_MANUAL command derives the default pointers from the physically next and prior screens in the source file. A nonexistent screen name has been temporarily used.

User Action: Add a DEFINE_PRIOR directive to the first screen in the source file.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_PRIOR directive.

**--WARNING HU 10-- define_up command must be supplied for the first screen.
Up set to point to {text}.**

Condition Identifier: HUE\$FIRST_UP_MUST_EXIST Code: HU 10

Description: The first screen in the source file must contain a DEFINE_UP directive because of the way the CREATE_MANUAL command derives the default pointers from the physically next and prior screens in the source file. A nonexistent screen name has been temporarily used.

User Action: Add a DEFINE_UP directive to the first screen in the source file.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_UP directive.

**--WARNING HU 15-- The last screen of a manual must have a define_next
command explicitly specified. Next set to point to {text}.**

Condition Identifier: HUE\$LAST_NEXT_MUST_EXIST Code: HU 15

Description: The last screen in the source file must contain a DEFINE_NEXT directive because of the way the CREATE_MANUAL command derives the default pointers from the physically next and prior screens in the source file. A nonexistent screen name has been temporarily used.

User Action: Add a DEFINE_NEXT directive to the last screen in the source file.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_NEXT directive.

**--WARNING HU 20-- Screen {text} contains more than one define_next command.
The last one was used.**

Condition Identifier: HUE\$EXTRA_NEXT_PTR Code: HU 20

Description: Only one DEFINE_NEXT directive can be specified on a screen.

User Action: Check the DEFINE_NEXT directives for the screen, change them as necessary so that you have the one you want, then delete any others.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_NEXT directive.

**--WARNING HU 25-- Screen {text} contains more than one define_prior
command. The last one was used.**

Condition Identifier: HUE\$EXTRA_PRIOR_PTR Code: HU 25

Description: Only one DEFINE_PRIOR directive can be specified on a screen.

User Action: Check the DEFINE_PRIOR directives for the screen, change them as necessary so that you have the one you want, then delete any others.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_PRIOR directive.

**--WARNING HU 30-- Screen {text} contains more than one define_up command.
The last one was used.**

Condition Identifier: HUE\$EXTRA_UP_PTR Code: HU 30

Description: Only one DEFINE_UP directive can be specified on a screen.

User Action: Check the DEFINE_UP directives for the screen, change them as necessary so that you have the one you want, then delete any others.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_UP directive.

--WARNING HU 35-- Screen {text} contains more than one menu/link definition for choice {text}. The first definition was used.

Condition Identifier: HUE\$DUPLICATE_MENU_ITEM Code: HU 35

Description: A menu letter cannot correspond to more than one selected screen or manual link.

User Action: Check the DEFINE_MENU and DEFINE_MANUAL_LINK directives on the screen and change them as necessary so that your menu correspond to one menu or manual link.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_MENU or DEFINE_MANUAL_LINK directives.

--WARNING HU 40-- Screen {text} contains more menu/link items than the maximum of {text} allowed in a single screen.

Condition Identifier: HUE\$EXCESS_MENU_ITEMS Code: HU 40

Description: Up to 36 menu items can be specified on a screen.

User Action: Reduce the number of DEFINE_MENU and DEFINE_MANUAL_LINK directives to 36 or fewer, then reduce the menu options on the screen as appropriate.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_MENU or DEFINE_MANUAL_LINK directive.

--WARNING HU 45-- Screen {text} contains no text lines.

Condition Identifier: HUE\$NUGATORY_SCREEN Code: HU 45

Description: This situation can occur when the source file contains a CREATE_SCREEN directive that you did not intend to have. The CREATE_SCREEN directive for the named screen might be an imperfect copy of the preceding or following CREATE_SCREEN directive source file line.

User Action: Either add text for the screen defined by the CREATE_SCREEN directive or delete the CREATE_SCREEN directive.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 50-- Screen {text} contains more than the {text} lines allowed. Excess lines may overflow the terminal screen.

Condition Identifier: HUE\$MAX_LINES_PER_SCREEN Code: HU 50

Description: Each screen can contain a maximum of 255 lines; however, only 24 lines can be displayed at a time on most terminal screens. The first 23 lines can contain text, while the last is used by the EXPLAIN prompt.

User Action: Reduce the number of lines of information on the screen to less than or equal to 23 lines. Unless the number of lines is considerably less than 255 or the SET_TERMINAL_ATTRIBUTE PAGE_HOLD=YES command is used, some lines probably cannot be read.

--WARNING HU 55-- Screen name {text} already used in this manual. Duplicate screen ignored. Screen {text} precedes the ignored screen in the input file.

Condition Identifier: HUE\$SCREEN_ALREADY_USED Code: HU 55

Description: Each screen must have a unique screen name.

User Action: Change the screen name (defined by the CREATE_SCREEN directive) for the duplicate screen.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 60-- In screen {text}, the maximum of {text} index entries at priority {text} for topic {text} has been exceeded. Excess entry not indexed.

Condition Identifier: HUE\$MAX_INDEX_ENTRIES_EXCEEDED Code: HU 60

Description: There are too many index entries for this subject with the same priority level.

User Action: Either delete the extra entries or, if you have not used all the priority levels for the subject, change its priority.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_INDEX_TOPIC directive.

--WARNING HU 65-- The output file you specified is an existing file containing data. It must be an empty new file.

Condition Identifier: HUE\$OUTPUT_FILE_EXISTS Code: HU 65

Description: Self-explanatory.

User Action: Reenter the CREATE_MANUAL command, but do one of the following:

Use the DELETE_FILE command first to discard the old bound manual.

Specify a new cycle of the current bound manual file in the CREATE_MANUAL command.

Specify a new empty file in the CREATE_MANUAL command.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_MANUAL command.

--WARNING HU 70-- The preceding create_screen command is invalid because of the following error (see below). Invalid create_screen command has been treated as a create_screen command with the name parameter omitted.

Condition Identifier: HUE\$INVALID_SCREEN_COMMAND Code: HU 70

Description: The directive indicated does not meet syntax requirements.

User Action: If you want the screen to have a specific name (so that you can refer to it in another directive), correct the CREATE_SCREEN directive so that it has a correct name parameter. Otherwise, omit the screen name, and EXPLAIN will provide a default screen name.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 75-- The preceding define_next command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_NEXT_COMMAND Code: HU 75

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_NEXT directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_NEXT directive.

--WARNING HU 80-- The preceding define_prior command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_PRIOR_COMMAND Code: HU 80

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_PRIOR directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_PRIOR directive.

--WARNING HU 85-- The preceding define_up command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_UP_COMMAND Code: HU 85

Description: The directive indicated does not meet syntax requirements. The default screen name (from the last DEFINE_UP directive) and static UP processing are used.

User Action: Correct the DEFINE_UP directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_UP directive.

--WARNING HU 90-- The preceding define_menu command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_MENU_COMMAND Code: HU 90

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_MENU directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_MENU directive.

--WARNING HU 95-- The preceding define_index command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_INDEX_COMMAND Code: HU 95

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_INDEX_TOPIC directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_INDEX_TOPIC directive.

--WARNING HU 100-- The preceding define_manual_link command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_LINK_COMMAND Code: HU 100

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_MANUAL_LINK directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_MANUAL_LINK directive.

--WARNING HU 105-- The preceding define_directive_char command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_DIRECTIVE_CHAR_CMD Code: HU 105

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_DIRECTIVE_CHAR directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_DIRECTIVE_CHAR directive.

--WARNING HU 110-- The create_screen command for the first screen is missing. Screen contents ignored.

Condition Identifier: HUE\$NO_INITIAL_SCREEN_COMMAND Code: HU 110

Description: One of the following events probably occurred:

More than 255 lines precede the first supplied CREATE_SCREEN directive.

A DEFINE_NEXT, DEFINE_PRIOR, DEFINE_UP, DEFINE_INDEX_TOPIC, DEFINE_MENU, or DEFINE_MANUAL_LINK directive or text lines precede the first supplied CREATE_SCREEN directive.

The file used as input to the CREATE_MANUAL command is not an online manual source file.

User Action: Add a CREATE_SCREEN directive to the first screen in the source file.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 115-- Screen {text} contains more than the maximum of {text} lines. Excess lines discarded.

Condition Identifier: HUE\$MAX_LINES Code: HU 115

Description: Each screen can contain a maximum of 255 lines (only 23 lines can be displayed at a time on most terminal screens).

User Action: Reduce the number of lines on the screen to 255 or fewer lines. Unless the lines on the screen are considerably less than 255 or the PAGE_HOLD terminal attribute is used, some lines probably cannot be read.

--WARNING HU 120-- This manual does not contain a screen named MAIN. Every manual must have a screen named MAIN because that is the name of the initial screen displayed when you read a manual.

Condition Identifier: HUE\$NO_MAIN_SCREEN Code: HU 120

Description: A spelling error might exist on the corresponding CREATE_SCREEN directive.

User Action: Add a screen named MAIN to the manual or rename the first screen to be displayed.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 125-- The preceding define_manual_name command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_DEFMN_COMMAND Code: HU 125

Description: The directive indicated does not meet SCL syntax requirements.

User Action: Correct the DEFINE_MANUAL_NAME directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_MANUAL_NAME directive.

--WARNING HU 130-- The preceding define_alternate_command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_DEFAC_COMMAND Code: HU 130

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_ALTERNATE_COMMAND directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_ALTERNATE_COMMAND directive.

--WARNING HU 135-- This manual does not contain a screen named HELP. Every manual should have a screen named HELP because that is the name of the screen displayed when the reader asks for help when no local help screen is defined.

Condition Identifier: HUE\$NO_HELP_SCREEN Code: HU 135

Description: EXPLAIN uses the default screen named HELP when the writer does not specify an alternate screen in a DEFINE_LOCAL_HELP_SCREEN directive.

User Action: Add a screen named HELP to the manual. HELP should explain the subcommands recognized by EXPLAIN within that manual.

Further Information: See the CONTEXT Usage Manual for information about the HELP screen.

--WARNING HU 140-- Screen {text} cannot be reached from any other screen in the manual.

Condition Identifier: HUE\$SCREEN_CANT_BE_REACHED Code: HU 140

Description: For a screen to be accessed from another screen in a manual, it must be indexed, be used in a DEFINE_MENU directive on some other screen, or be the prior or next screen of some other screen.

User Action: Decide how you want the screen to be accessed and add the necessary directives to reference it.

Further Information: See the CONTEXT Usage Manual for information about using directives to reference screens.

--WARNING HU 145-- Screen {text} is referenced but was never created.

Condition Identifier: HUE\$SCREEN_REF_BUT_NOT_DEF Code: HU 145

Description: Screens are referenced by being named in one or more of the directives DEFINE_MENU, DEFINE_UP, DEFINE_NEXT, DEFINE_PRIOR, and DEFINE_INDEX_TOPIC.

User Action: Create the screen or delete references to it.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 147-- Screen {text} is referenced but was never created. The reference occurs in a {text} command from screen {text}.

Condition Identifier: HUE\$SCREEN_REF_BUT_NOT_DEF_FROM Code: HU 147

Description: The screen name at the beginning of the message appears in the embedded subcommand indicated. That subcommand appears in the screen named at the end of the message. The first screen name either contains a typographical error or no CREATE_SCREEN directive exists for the appropriate screen.

User Action: Create the screen or delete references to it.

Further Information: See the CONTEXT Usage Manual for information about the CREATE_SCREEN directive.

--WARNING HU 150-- The preceding define_scl_procedure command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_PROC_COMMAND Code: HU 150

Description: The directive indicated does not meet syntax requirements. All text lines following the ignored directive will display.

User Action: Correct the DEFINE_SCL_PROCEDURE directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_SCL_PROCEDURE directive.

--WARNING HU 155-- The preceding delete_scl_procedure command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_DELETE_COMMAND Code: HU 155

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the DEFINE_SCL_PROCEDURE directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_SCL_PROCEDURE directive.

--WARNING HU 160-- The preceding execute_scl_procedure command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_EXECUTE_COMMAND Code: HU 160

Description: The directive indicated does not meet syntax requirements.

User Action: Correct the EXECUTE_SCL_PROCEDURE directive.

Further Information: See the CONTEXT Usage Manual for information about the EXECUTE_SCL_PROCEDURE directive.

--WARNING HU 165-- The preceding define_scl_procedure_end command was ignored because of the following error (see below).

Condition Identifier: HUE\$INVALID_PROCEND_COMMAND Code: HU 165

Description: The directive indicated does not meet syntax requirements. The procedure file will include all text lines until the next CREATE_SCREEN directive.

User Action: Correct the DEFINE_SCL_PROCEDURE_END directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_SCL_PROCEDURE_END directive.

--WARNING HU 167-- The preceding define_local_help_screen command is invalid because of the following error (see below). Invalid define_local_help_screen command has been treated as a define_local_help_screen command with the name parameter omitted.

Condition Identifier: HUE\$INVALID_HELP_COMMAND Code: HU 167

Description: The directive indicated does not meet syntax requirements. The default screen name of HELP was used.

User Action: Correct the DEFINE_LOCAL_HELP directive.

Further Information: See the CONTEXT Usage Manual for information about the DEFINE_LOCAL_HELP_SCREEN directive.

--WARNING HU 168-- The last line of screen {text} contains more than 79 characters. This can cause some terminals to scroll part of the display off screen using line mode explain.

Condition Identifier: HUE\$LONG_LINE_SCROLLS_TERMINAL Code: HU 168

Description: The last line of text on your screen contains more than 79 characters. If the line is longer than 79 characters, any text past the 79th character is lost.

User Action: Ensure the line of text contains the maximum of 79 characters.

--ERROR HU 500-- {text} is not a manual.

Condition Identifier: HUE\$FILE_IS_NOT_A_MANUAL Code: HU 500

Description: What you entered is not the name of a manual.

User Action: Enter the name of the manual you want to read.

--INFORMATIVE HU 501-- Please press DATA and select a topic if you wish to do a topic search.

Condition Identifier: HUE\$ENTER_QUERY_721 Code: HU 501

Description: Self-explanatory.

User Action: To use the index, press the DATA key, type the subject that you want to search for, then press the NEXT key.

--INFORMATIVE HU 505-- Attempted to access missing screen {text}. This is an error in the writing of the manual and should be reported to the author.

Condition Identifier: HUE\$MISSING_SCREEN Code: HU 505

Description: The manual uses the indicated screen name as a pointer but the manual contains no screen by that name. The writer also did not include a screen named \$NO_SCREEN to explain this problem to the reader. When the manual was processed by the CREATE_MANUAL command, one of the following events occurred:

The writer omitted a required directive and a temporary but invalid default screen name was used.

The writer misspelled a valid screen name in a directive.

The writer forgot to provide a screen.

User Action: The reader cannot correct the problem, but can still use the book. Type UP, TOP, or minus (-), or press RETURN to go to another part of the manual. If CDC wrote the manual, please report the error to CDC Publications and Graphics Division.

Further Information: If CDC wrote the manual, please report the steps that led you to this screen to the address indicated within the manual under the topic SUBMITTING COMMENTS. Follow the instructions given under that topic.

--INFORMATIVE HU 506-- Attempted to access a missing screen from the screen {text}. This is an error in the writing of the manual and should be reported.

Condition Identifier: HUE\$MISSING_SCREEN_FROM Code: HU 506

Description: The manual uses an invalid screen name as a pointer from the screen with the indicated name.

The writer also did not include a screen named \$NO_SCREEN to explain this problem to the reader.

When the manual was processed by the CREATE_MANUAL command, one of the following events occurred:

The writer omitted a required directive and a temporary but invalid default screen name was used.

The writer misspelled a valid screen name in a directive.

The writer forgot to provide a screen.

User Action: The reader cannot correct the problem, but can still use the book. Type UP, TOP, or minus (-), or press RETURN to go to another part of the manual. If CDC wrote the manual, please report the error to CDC Publications and Graphics Division.

Further Information: If CDC wrote the manual, please report the steps that led you to this screen to the address indicated within the manual under the topic SUBMITTING COMMENTS. Follow the instructions given under that topic.

--INFORMATIVE HU 507-- Attempted to access a missing screen. This is an error in the writing of the manual and should be reported.

Condition Identifier: HUE\$MISSING_SCREEN_FROM_NOWHERE Code: HU 507

Description: The manual uses an invalid screen name as a pointer. The writer also did not include a screen named \$NO_SCREEN to explain this problem to the reader. When the manual was processed by the CREATE_MANUAL command, one of the following events occurred:

The writer omitted a required directive and a temporary but invalid default screen name was used.

The writer misspelled a valid screen name in a directive.

The writer forgot to provide a screen.

User Action: The reader cannot correct the problem, but can still use the book. Type UP, TOP, or minus (-), or press RETURN to go to another part of the manual. If CDC wrote the manual, please report the error to CDC Publications and Graphics Division.

Further Information: If CDC wrote the manual, please report the steps that led you to this screen to the address indicated within the manual under the topic SUBMITTING COMMENTS. Follow the instructions given under that topic.

--INFORMATIVE HU 510-- Please enter topic? if you wish to do a ? search.

Condition Identifier: HUE\$ENTER_QUERY Code: HU 510

Description: Self-explanatory.

User Action: To use the index, type a subject that you want to search for, followed by a ?, and then press RETURN.

--INFORMATIVE HU 515-- Could not find anything for the topic - {text}.

Condition Identifier: HUE\$COULD_NOT_FIND_TOPIC Code: HU 515

Description: No close match exists for the subject you tried to reach through the index.

User Action: Follow the instructions in the message or use the DISI subcommand to look for possible alternate indexed subjects.

Further Information: See the CONTEXT Usage Manual for information about using an online manual index.

--INFORMATIVE HU 520-- Topic entries exhausted for - {text}.

Condition Identifier: HUE\$TOPICS_EXHAUSTED Code: HU 520

Description: Self-explanatory.

User Action: Use any of the subcommands for reading a manual.

Further Information: See the CONTEXT Usage Manual for information about using an online manual index.

--INFORMATIVE HU 523-- Press SHIFT and STOP together to leave EXPLAIN.

Condition Identifier: HUE\$ENTER_SHIFT_STOP Code: HU 523

Description: Self-explanatory.

User Action: To leave EXPLAIN, hold down the SHIFT key and press the STOP key.

Further Information: See the CONTEXT Usage Manual for information about the QUIT (SHIFT-STOP) subcommand.

--INFORMATIVE HU 525-- Enter QUIT to leave EXPLAIN.

Condition Identifier: HUE\$ENTER_QUIT Code: HU 525

Description: Self-explanatory.

User Action: To leave EXPLAIN, type QUIT and press RETURN.

Further Information: See the CONTEXT Usage Manual for information about the QUIT subcommand.

--INFORMATIVE HU 530-- The manual you specified, {file}, could not be found.

Condition Identifier: HUE\$MANUAL_NOT_FOUND Code: HU 530

Description: Either the file name used is wrong, or the manual corresponding to the named file is not available on your system.

User Action: Enter DISPLAY_CATALOG \$SYSTEM.MANUALS or go to the online manual access screen in the NOS/VE System Information Manual and from there access other available manuals.

--INFORMATIVE HU 532-- EXPLAIN cannot link to the manual, {file}. It is not installed on your system.

Condition Identifier: HUE\$LINK_NOT_FOUND Code: HU 532

Description: EXPLAIN cannot link to the manual file named. Either the file name is wrong, or the manual corresponding to that file is not available on your system.

User Action: The writer must check the file name used and change the manual containing the link. The reader cannot do anything to fix this problem.

--INFORMATIVE HU 535-- {text} screens excerpted on to the listing file.

Condition Identifier: HUE\$SCREENS_EXCERPTED Code: HU 535

Description: The screen displayed above this message is the last screen copied.

User Action: If you want a printed copy of the screens written to the file, print the file.

Further Information: See the CONTEXT Usage Manual for information about the EXCERPT subcommand.

--INFORMATIVE HU 540-- You are in the original manual {file}. There is no manual to revert to.

Condition Identifier: HUE\$NO_LINK_EXISTS Code: HU 540

Description: The REVERT subcommand has no effect unless you enter one manual from another manual via a menu item the writer of the original manual supplied.

User Action: Use QUIT to exit the current manual.

Further Information: See the CONTEXT Usage Manual for information about the REVERT subcommand.

--INFORMATIVE HU 545-- You have used the undocumented \$child parameter or there has been an error in task termination.

Condition Identifier: HUE\$USER_SAID_QUIT Code: HU 545

Description: The \$CHILD parameter is reserved for use by CDC programming environments.

User Action: No current action necessary. Do not use the \$CHILD parameter for the EXPLAIN command.

--INFORMATIVE HU 550-- {text} is not a command. Press FWD for the next screen, BKW for the previous screen, SHIFT-BACK to go to the start of the manual or SHIFT-CLEAR to repaint the current screen. You may also type DISI to view a menu of index topics or press HELP for further assistance.

Condition Identifier: HUE\$NO_UNDERSTAND_721 Code: HU 550

Description: If you tried to make a menu selection, the writer did not supply any DEFINE_MENU directive for this screen. You also might have pressed a function key that EXPLAIN does not support. EXPLAIN thinks you entered the characters shown to use a subcommand it does not support.

User Action: Follow the instructions in the message. You can also use SHIFT-CLEAR or DATA.

Further Information: See the CONTEXT Usage Manual for information about the EXPLAIN utility subcommands for the CDC 721 terminal.

--INFORMATIVE HU 555-- {text} is not a command. Press RETURN for the next screen or minus (-) for the previous screen. Type TOP to go to the start of the manual, REFRESH to repaint the current screen, DISI to view a menu of index topics or HELP for further assistance.

Condition Identifier: HUE\$NO_UNDERSTAND Code: HU 555

Description: If you tried to make a menu selection, the writer did not supply any DEFINE_MENU directive for this screen. You also might have entered a find indexed subject subcommand (xxxx?) without a question mark at the end. EXPLAIN thinks you tried to enter a subcommand it does not recognize.

User Action: Follow the instructions in the message or reenter the xxxx? subcommand. You can also use the REFRESH or DISI subcommands.

Further Information: See the CONTEXT Usage Manual for information about the EXPLAIN utility subcommands.

--INFORMATIVE HU 560-- {text} is not a permitted entry here. Make a menu selection, press FWD for the next screen, BKW for the previous screen, SHIFT-BACK to go to the start of the manual or SHIFT-CLEAR to repaint the current screen. You may also type DISI to view a menu of index topics or press HELP for further assistance.

Condition Identifier: HUE\$NO_UNDERSTAND_MENU_721 Code: HU 560

Description: The writer might have made a typographical error in the displayed menu text. The screen has no DEFINE_MENU directive for the characters you entered.

User Action: Follow the instructions in the message or report the problem to the manual writer.

Further Information: See the CONTEXT Usage Manual for information about using an online manual menu.

--INFORMATIVE HU 565-- {text} is not a permitted entry here. Make a menu selection, press RETURN for the next screen or minus (-) for the previous screen. You may also type TOP to go to the start of the manual, REFRESH to repaint the current screen, DISI to view a menu of index topics or HELP for further assistance.

Condition Identifier: HUE\$NO_UNDERSTAND_MENU Code: HU 565

Description: The writer might have made a typographical error in the displayed menu text. The screen has no DEFINE_MENU directive for the characters you entered.

User Action: Follow the instructions in the message or report the problem to the manual writer.

Further Information: See the CONTEXT Usage Manual for information about using an online manual index.

--INFORMATIVE HU 570-- < What topic >

Condition Identifier: HUE\$WHAT_TOPIC Code: HU 570

Description: The phrase What Topic is a prompt that appears when the DATA key is pressed on a CDC 721 terminal. Pressing the DATA key is the first step in beginning an indexed subject search.

User Action: Enter an indexed subject.

Further Information: See the CONTEXT Usage manual for information about using an online manual index.

--INFORMATIVE HU 575-- No index topics exist for this manual.

Condition Identifier: HUE\$NO_INDEX_ENTRIES_EXIST Code: HU 575

Description: The writer provided no DEFINE_INDEX_TOPIC directives in the book, and the reader has tried to display the index using the DISI command.

User Action: None. The message is informative only.

--INFORMATIVE HU 580-- {text} is not a permitted entry here. Make a menu selection, press FWD or BKW for more index topics, HELP for assistance, or BACK to return to the screen from which the disI command was entered.

Condition Identifier: HUE\$NO_UNDERSTAND_MENU_DISI_721 Code: HU 580

Description: You cannot use the command or key shown on a dynamically generated menu. The DISI menus are created as you call for them and do not actually appear in the manual as text the writer provided.

User Action: Follow the instructions in the message.

--INFORMATIVE HU 585-- {text} is not a permitted entry here. Make a menu selection, press RETURN or - for more index topics, type HELP for assistance, or BACK to return to the screen from which the disI command was entered.

Condition Identifier: HUE\$NO_UNDERSTAND_MENU_DISI Code: HU 585

Description: You cannot use the command shown when on a dynamically generated menu. The DISI menus are created as you call for them and do not actually appear in the manual as text the writer provided.

User Action: Follow the instructions in the message.

--INFORMATIVE HU 590-- Embedded SCL proc encountered for this screen.

Condition Identifier: HUE\$EMBEDDED_SCL_PROC_MESSAGE1 Code: HU 590

Description: You are being warned that the current screen of the online manual contains an embedded System Command Language procedure file. This warning is issued only for manuals that are not explicitly your own or installed by your site.

User Action: As the messages indicates, you can inspect the file to ensure that it contains no commands which might destroy your files or otherwise compromise system security. Unless you know what the file contains, you should inspect it or continue without executing it.

Further Information: See the CONTEXT Usage Manual for information about embedded SCL procedures.

--ERROR HU 800-- The application menu {text} contains too many menu items. It must be corrected prior to using EXPLAIN.

Condition Identifier: HUE\$MENU_TOO_LARGE Code: HU 800

Description: The menu definition in the message module HUM\$US_ENGLISH is incorrect.

User Action: Correct the menu definition.

--WARNING HU 815-- Some lines on this and other screens may be obscured by the function key display. This message will not be displayed again. You can press {text} to remove the key display.

Condition Identifier: HUE\$SOME_OBSCURED_LINES Code: HU 815

Description: When using a manual in screen mode, there may be some lines on a screen that will be covered by the function keys.

User Action: None.

--ERROR IC 0-- File {file} : The link to NOS/170 is already open.

Condition Identifier: ICE\$LINK_IS_ALREADY_OPEN Code: IC 0

Description: The job already has a link file open.

User Action: Wait until the current link file is closed before attempting to open another link file.

--ERROR IC 5-- File {file} : User_Info is empty or does not name a defined SCL string variable.

Condition Identifier: ICE\$NO_JOB_SPEC_VARIABLE Code: IC 5

Description: The user_info attribute of the link file must contain the name of an SCL string variable. The SCL variable contains control statements for the NOS job.

User Action: Set the user_info attribute value using a SET_FILE_ATTRIBUTES command or AMP\$FILE or AMP\$OPEN call.

--ERROR IC 10-- File {file} : The SCL string variable {text} must contain a NOS/170 job deck, but is empty.

Condition Identifier: ICE\$EMPTY_JOB_SPEC_VARIABLE Code: IC 10

Description: The command variable specified by the link file user_info attribute is empty.

User Action: Store all control statements for the NOS job except the CHARGE and USER statements in the command variable. (The system generates the USER and CHARGE statements.)

--ERROR IC 15-- File {file} : The control statements for the partner NOS/170 job exceed the allowed length, beginning with statement number {text}.

Condition Identifier: ICE\$PARTNER_JOB_TOO_LONG Code: IC 15

Description: The command variable containing the NOS control statements is too long to be transferred to NOS.

User Action: Store the NOS control statements as a NOS procedure file. Therefore, the only statements needed within the command variable are those required to call and execute the procedure.

--ERROR IC 20-- File {file} : The partner NOS/170 job cannot be started by NOS/170.

Condition Identifier: ICE\$PARTNER_CANNOT_BE_STARTED Code: IC 20

Description: The NOS job could not be started due to a job error such as lack of valid user identification information. The NOS accounting information must be specified using the SET_LINK_ATTRIBUTES or the CHANGE_LINK_ATTRIBUTES command within the NOS/VE job.

User Action: Ensure the valid NOS accounting information has been specified.

--ERROR IC 25-- File {file} : The link file must have access_level of RECORD, but it specifies {text}.

Condition Identifier: IC\$ACCESS_LEVEL_NOT_RECORD Code: IC 25

Description: The task attempted to open a link file for segment access.

User Action: Change the access level specification on the open call to record access.

--ERROR IC 30-- File {file} : This task and its partner NOS/170 job are both attempting PUTs. ({text} request.)

Condition Identifier: IC\$WRITE_DEADLOCK Code: IC 30

Description: The NOS job and NOS/VE task cannot both write to the link file at the same time.

User Action: Issue a get request to clear the write deadlock condition.

--ERROR IC 35-- File {file} : This task's partner NOS/170 job has ended or called CLOSLNK. ({text} request.)

Condition Identifier: IC\$PARTNER_ENDED Code: IC 35

Description: The NOS job has ended or requested that the link be closed.

User Action: Close the link file.

--ERROR IC 40-- File {file} : This task and its partner NOS/170 job are both attempting GETs. ({text} request.)

Condition Identifier: IC\$READ_DEADLOCK Code: IC 40

Description: The NOS job and NOS/VE task cannot both read from the link file at the same time.

User Action: Issue a put request to clear the read deadlock condition.

--ERROR IC 45-- The partner NOS/170 job cannot be run due to an unexpected internal system status {text}.

Condition Identifier: IC\$UNEXPECTED_ML_ERROR Code: IC 45

Description: An unexpected memory link error has occurred.

User Action: Notify your site analyst.

--ERROR IC 50-- The dual state code for the NOS/VE and {text} sides do not match. Contact your site analyst.

Condition Identifier: IC\$MISMATCHING_CODE Code: IC 50

Description: The NOS dual state binaries do not match what is running on NOS/VE. Therefore, dual state commands do not work.

User Action: Contact your site analyst.

--ERROR IC 55-- The {text} partner job is already signed on to the memory link but by a different application name. Retry the command or contact your site analyst.

Condition Identifier: IC\$RECEIVER_ALREADY_SIGNED_ON Code: IC 55

Description: The dual state partner job is trying to sign on to the memory link, but is already signed on by a different application name.

User Action: Try the command again or contact your site analyst.

--ERROR IC 60-- A problem occurred on the {text} side - check your {text} validations or your {text} file being busy.

Condition Identifier: IC\$RECEIVER_NOT_SIGNED_ON Code: IC 60

Description: The dual state partner job could not sign on to the memory link due either to incorrect 170 validations or the 170 file being busy.

User Action: Verify your validations on the 170 and verify that the 170 file is not busy. If the problem persists, contact your site analyst.

--ERROR IC 65-- This system is not running NOS or NOS/BE dual state, so the interstate facility is not available.

Condition Identifier: ICE\$NO_PARTNER_EXISTS Code: IC 65

Description: A command to use the Interstate Facility was issued by a user on a NOS/VE standalone system. The system is running NOS/VE only. Interstate communication facility commands are not available in non-dual state systems.

User Action: None.

--ERROR IF 5-- File {file} must be assigned to a terminal device.

Condition Identifier: IF\$FILE_NAME_NOT_TERMINAL Code: IF 5

Description: The specified local file name is assigned to a device class other than the terminal device class.

User Action: Assign the local file name to the terminal device class.

--ERROR IF 10-- File {text} cannot be found.

Condition Identifier: IF\$FILE_NAME_NOT_FOUND Code: IF 10

Description: The specified local file name is not defined within the job.

User Action: Ensure that an RMP\$REQUEST_TERMINAL call assigns the local file name to the terminal device class.

--ERROR IF 15-- File name {text} is ill-formed.

Condition Identifier: IF\$FILE_NAME_ILL_FORMED Code: IF 15

Description: The local file name is specified incorrectly.

User Action: Correct the local file name.

--INFORMATIVE IF 20-- The current job must be of interactive mode.

Condition Identifier: IF\$CURRENT_JOB_NOT_INTERACTIVE Code: IF 20

Description: The system could not perform the request because the job did not originate at an interactive terminal.

User Action: Log in to a terminal and execute the task.

--ERROR IF 25-- The current job must not be disconnected.

Condition Identifier: IF\$CURRENT_JOB_DISCONNECTED Code: IF 25

Description: The current job must not be disconnected.

User Action: None.

--ERROR IF 30-- Job name {text} is ill-formed.

Condition Identifier: IF\$JOB_NAME_ILL_FORMED Code: IF 30

Description:

User Action: Not applicable.

--ERROR IF 35-- Job {text} cannot be found.

Condition Identifier: IF\$JOB_NOT_FOUND Code: IF 35

Description: Diagnostic description currently not implemented.

NOTE

This message is currently undocumented because it is for an internal, unsupported, or new interface. Unless an actual diagnostic description is supplied, this message will be automatically excluded from the final Diagnostic Messages manual.

User Action: Not applicable.

--ERROR IF 40-- Job {text} must be a disconnected interactive job.

Condition Identifier: IF\$OTHER_JOB_NOT_DISCONNECTED Code: IF 40

Description: This type of job must be a disconnected interactive job.

User Action: Correct the job name and resubmit.

--ERROR IF 70-- Advance unit {text} is unknown.

Condition Identifier: IF\$UNKNOWN_ADVANCE_UNIT Code: IF 70

Description: The user has specified an unknown advance unit on a IFP\$ADVANCE request.

User Action: Specify an advance unit which is of type IFT\$ADVANCE_UNIT and reissue the request.

--ERROR IF 75-- Terminal attribute key {text} is unknown.

Condition Identifier: IF\$UNKNOWN_ATTRIBUTE_KEY Code: IF 75

Description: A key field in the terminal attribute array contains an invalid identifier. Each terminal attribute identifier begins with the characters IFC\$ followed by the attribute name.

User Action: Correct the key field value.

--ERROR IF 80-- Terminal attribute key {text} is invalid for this request.

Condition Identifier: IF\$INVALID_KEY_FOR_REQUEST Code: IF 80

Description: The call cannot specify a value for the terminal attribute.

User Action: If possible, assign the terminal attribute value with a SET_TERMINAL_ATTRIBUTE command or IFP\$CHANGE_TERMINAL_ATTRIBUTES call.

--ERROR IF 85-- Terminal attribute {text} of value '{text}' must not be the same as attribute {text}.

Condition Identifier: IF\$CONTROL_CHAR_CONFLICT Code: IF 85

Description: The values of the two attributes must not be the same.

User Action: Change one of the attributes.

--ERROR IF 90-- For {text}, {text} is outside the range of legal values.

Condition Identifier: IF\$ATTR_VAL_DISALLOWED_BY_NAM Code: IF 90

Description: This terminal attribute value is not allowed by NAM.

User Action: Change the terminal attribute value.

--INFORMATIVE IF 91-- PBC, TC and TBC are read-only or non-applicable on a NAM/VE network.

Condition Identifier: IF\$ILLEGAL_NAM_VE_CHANGE Code: IF 91

Description: Pause_Break_Character, Terminate_Break_Character and/or Terminal_Class have been specified on a request to change the terminal attributes on a NAM/VE network, these attributes are non-applicable. Any remaining attribute changes on the request will be honored.

User Action: None

--INFORMATIVE IF 92-- PBC and TBC are read-only attributes on a NAM/CDCNET network.

Condition Identifier: IF\$ILLEGAL_NAM_CDCNET_CHANGE Code: IF 92

Description: Pause_Break_Character and/or Terminate_Break_Character have been specified on a request to change the terminal attributes on a NAM/CDCNET network, for which these attributes are read-only. Any remaining attribute changes on the request will be honored.

User Action: None.

--INFORMATIVE IF 93-- AC, BLC, CRS, CS, EPA, FFD,FFS, HPO, LFS, and EOS are read-only or non-applicable on a NAM/CCP network.

Condition Identifier: IFE\$ILLEGAL_NAM_CCP_CHANGE Code: IF 93

Description: Attention_Character, Begin_Line_Character, Carriage_Return_Sequence, Code_Set, End_Page_Action, Form_Feed_Delay, Form_Feed_Sequence, Hold_Page_Over, Line_Feed_Sequence and/or End_Output_Sequence have been specified on a request to change the terminal attributes on a NAM/CCP network, these attributes are non-applicable or read-only. Any remaining attribute changes on the request will be honored.

User Action: None.

--INFORMATIVE IF 94-- TPM is non-applicable on a NAM/CCP network.

Condition Identifier: IFE\$ILLEGAL_NAM_CCP_CONN_CHANGE Code: IF 94

Description: Transparent_Protocol_Mode has been specified on a request to change the terminal connection attributes on a NAM/CCP network, this attribute is non-applicable. Any remaining attribute changes on the request will be honored.

User Action: None.

--ERROR IF 95-- Carriage return delay value {text} must be from {text} to {text}.

Condition Identifier: IFE\$CR_DELAY Code: IF 95

Description: The user has attempted to change the Carriage_Return_Delay terminal attribute and has specified a delay value that is outside the range of values defined by IFT\$CARRIAGE_RETURN_DELAY.

User Action: Specify a Carriage_Return_Delay value that is within IFT\$CARRIAGE_RETURN_DELAY (currently 0 to 999) and reissue the terminal attribute change.

--ERROR IF 110-- End Of Information string '{text}' size must be from {text} to {text} characters.

Condition Identifier: IFE\$END_OF_INFORMATION_SIZE Code: IF 110

Description: You have attempted to change the End_Of_Information string connection attribute and have specified a string whose length is outside the range of values defined by IFT\$END_OF_INFORMATION_SIZE.

User Action: Specify an End_Of_Information string whose length is within IFT\$END_OF_INFORMATION_SIZE and reissue the connection attribute change.

--ERROR IF 125-- Input Output Mode {text} must be either: solicited, unsolicited, or full_duplex.

Condition Identifier: IFE\$UNKNOWN_INPUT_OUTPUT_MODE Code: IF 125

Description: You have selected an Input_Output_Mode value which is not a valid option for this connection attribute.

User Action: Specify an Input_Output_Mode option of unsolicited, solicited, or full duplex and reissue the connection attribute change.

--ERROR IF 126-- Status Action {text} must be either: send, hold or discard.

Condition Identifier: IFE\$UNKNOWN_STATUS_ACTION Code: IF 126

Description: You have selected a Status_Action value which is not a valid option for this terminal attribute change.

User Action: Specify a Status_Action option of send, hold, or discard and reissue the terminal attribute change.

--ERROR IF 127-- End Partial Positioning {text} must be either: none, crs, lfs, or crslfs.

Condition Identifier: IFE\$UNKNOWN_END_PART_POSITION Code: IF 127

Description: The user has selected an End_Partial_Positioning option which is not valid for this terminal attribute change.

User Action: Specify an End_Partial_Positioning option of none, crs, lfs, or crslfs and reissue the terminal attribute change.

--ERROR IF 130-- Line feed delay value {text} must be from {text} to {text}.

Condition Identifier: IFESLF_DELAY Code: IF 130

Description: The user has attempted to change the Line_Feed_Delay terminal attribute and has specified a delay value that is outside the range of values defined by IFT\$LINE_FEED_DELAY.

User Action: Specify a Line_Feed_Delay value that is within IFT\$LINE_FEED_DELAY (currently 0 to 999) and reissue the terminal attribute change.

--ERROR IF 135-- Output device {text} must be either display or printer.

Condition Identifier: IFESUNKNOWN_OUTPUT_DEVICE Code: IF 135

Description: The valid OUTPUT_DEVICE attribute values are AMC\$DISPLAY_OUTPUT, AMC\$PRINTER_OUTPUT, and AMC\$PAPER_TAPE_OUTPUT.

User Action: Correct the input_device attribute value.

--ERROR IF 140-- Page length {text} must be from {text} to {text} and not 1 on NAM/CDCNET.

Condition Identifier: IFESPAGE_LENGTH Code: IF 140

Description: The user has attempted to change the Page_Length terminal attribute and has specified a value that is outside the range of values defined by IFT\$PAGE_LENGTH.

User Action: Specify a Page_Length value that is within IFT\$PAGE_LENGTH and reissue the terminal attribute change.

--ERROR IF 145-- Page width {text} must be from {text} to {text} and not 1 to 9 on NAM/CDCNET.

Condition Identifier: IFESPAGE_WIDTH Code: IF 145

Description: The user has attempted to change the Page_Width terminal attribute and has specified a value that is outside the range of values defined by IFT\$PAGE_WIDTH.

User Action: Specify a Page_Width value that is within IFT\$PAGE_WIDTH and reissue the terminal attribute change.

--ERROR IF 150-- Parity mode {text} must be either: zero, mark, even, odd or none.

Condition Identifier: IFESUNKNOWN_PARITY_MODE Code: IF 150

Description: The specified parity mode must be NONE, EVEN, ODD, MARK, or ZERO.

User Action: Specify a valid parity mode.

--ERROR IF 155-- Prompt file name {text} is ill-formed.

Condition Identifier: IFESPROMPT_FILE_NAME_ILL_FORMED Code: IF 155

Description: The local file name is specified incorrectly.

User Action: Correct the specified local file name.

--ERROR IF 160-- Prompt file {text} cannot be found.

Condition Identifier: IFESPROMPT_FILE_NAME_NOT_FOUND Code: IF 160

Description: The specified local file name is not defined within the job.

User Action: Either correct the specified local file name or define the name within the job.

--ERROR IF 165-- Prompt file {text} must be assigned to a terminal device.

Condition Identifier: IFESPROMPT_FILE_NAME_NOT_TERM Code: IF 165

Description: The specified prompt file is not assigned to the terminal device class.

User Action: Issue an RMP\$REQUEST_TERMINAL interface call or execute the REQUEST_TERMINAL command to assign the local file name to the terminal device class.

--ERROR IF 170-- Prompt file {text} for prompt file ID cannot be found.

Condition Identifier: IFE\$PROMPT_FILE_ID_NOT_FOUND Code: IF 170

Description: The file identifier does not specify a currently open file.

User Action: Correct the prompt file identifier value.

--ERROR IF 175-- Prompt file {text} for prompt file ID must be assigned to a terminal device.

Condition Identifier: IFE\$PROMPT_FILE_ID_NOT_TERM Code: IF 175

Description: The prompt file identifier does not specify a terminal file.

User Action: Either correct the prompt file identifier or change the task so that the prompt file is assigned to the terminal device class.

--ERROR IF 180-- Prompt string '{text}' size must be from {text} to {text} characters.

Condition Identifier: IFE\$PROMPT_STRING_SIZE Code: IF 180

Description: The user has attempted to change the Prompt_String connection attribute and has specified a string whose length is outside the range of values defined by IFT\$PROMPT_STRING_SIZE.

User Action: Specify a Prompt_String whose length is within IFT\$PROMPT_STRING_SIZE (currently 0 to 31) and reissue the connection attribute change.

--ERROR IF 185-- Terminal class {text} is unknown.

Condition Identifier: IFE\$UNKNOWN_TERMINAL_CLASS Code: IF 185

Description: The specified terminal class is unknown.

User Action: Specify a valid terminal class.

--ERROR IF 190-- Trans Message Length {text} must be from {text} to {text}.

Condition Identifier: IFE\$TRANS_MESSAGE_LENGTH Code: IF 190

Description: The user has attempted to change the Trans_Message_Length connection attribute and has specified a value that is outside the range of values defined by IFT\$TRANS_MESSAGE_LENGTH.

User Action: Specify a Trans_Message_Length value that is within IFT\$TRANS_MESSAGE_LENGTH (currently 1 to 32767) and reissue the connection attribute change.

--ERROR IF 225-- Echoplex must be TRUE or FALSE.

Condition Identifier: IFE\$ECHOPLEX Code: IF 225

Description: The user has attempted to change the Echoplex terminal attribute and has specified a non-boolean value.

User Action: Specify an Echoplex value of TRUE or FALSE and reissue the terminal attribute change.

--ERROR IF 226-- Fold Line must be TRUE or FALSE.

Condition Identifier: IFE\$FOLD_LINE Code: IF 226

Description: The user has attempted to change the Fold_Line terminal attribute and has specified a non-boolean value.

User Action: Specify a Fold_Line value of TRUE or FALSE and reissue the terminal attribute change.

--ERROR IF 230-- Partial Char Forwarding must be TRUE or FALSE.

Condition Identifier: IFE\$PARTIAL_CHAR_FORWARDING Code: IF 230

Description: The user has attempted to change the Partial_Character_Forwarding connection attribute and has specified a non-boolean value.

User Action: Specify a Partial_Char_Forwarding value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 231-- Store Backspace Character must be TRUE or FALSE.

Condition Identifier: IF\$STORE_BACKSPACE_CHARACTER Code: IF 231

Description: The user has attempted to change the Store_Backspace_Character connection attribute and has specified a non-boolean value.

User Action: Specify a Store_Backspace_Character value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 232-- Store Nuls Dels must be TRUE or FALSE.

Condition Identifier: IF\$STORE_NULS_DELS Code: IF 232

Description: The user has attempted to change the Store_Nuls_Dels connection attribute and has specified a non-boolean value.

User Action: Specify a Store_Nuls_Dels value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 235-- Hold page must be TRUE or FALSE.

Condition Identifier: IF\$HOLD_PAGE Code: IF 235

Description: The user has attempted to change the Hold_Page terminal attribute and has specified a non-boolean value.

User Action: Specify a Hold_Page value of TRUE or FALSE and reissue the terminal attribute change.

--ERROR IF 270-- Pause break received from terminal.

Condition Identifier: IF\$PAUSE_BREAK_RECEIVED Code: IF 270

Description: A pause break has been received from the terminal.

User Action: Any command may be issued, plus RESUME_COMMAND or TERMINATE_COMMAND.

--ERROR IF 275-- Terminate break received from terminal.

Condition Identifier: IF\$TERMINATE_BREAK_RECEIVED Code: IF 275

Description: A terminate break has been received from the terminal.

User Action: None.

--INFORMATIVE IF 280-- Terminal disconnected from job due to broken terminal connection.

Condition Identifier: IF\$CONNECTION_BREAK_DISCONNECT Code: IF 280

Description: The job's physical connection to the terminal has been interrupted, therefore suspending the job.

User Action: After LOGIN the suspended job can either be terminated or re-attached and execution resumed.

--INFORMATIVE IF 295-- Terminal has been reconnected to this disconnected job.

Condition Identifier: IF\$TERMINAL_RECONNECTED_TO_JOB Code: IF 295

Description: The terminal has been reconnected to this disconnected job. The job is in a pause break state.

User Action: Any command may be issued, plus RESUME_COMMAND or TERMINATE_COMMAND.

--INFORMATIVE IF 300-- Terminal has been reconnected to a disconnected job -this job is being terminated.

Condition Identifier: IF\$TERMINAL_RECONNECTED_OTHER Code: IF 300

Description: The terminal has been reconnected to a disconnected job and this job is being terminated.

User Action: None.

--ERROR IF 305-- This job is terminating because it remained disconnected longer than the system allows.

Condition Identifier: IF\$DISCONNECTED_JOB_TIMEOUT Code: IF 305

Description: This job is terminating because it remained disconnected longer than the system allows.

User Action: The job must be rerun.

--ERROR IF 315-- Character flow control must be TRUE or FALSE.

Condition Identifier: IFES\$CHARACTER_FLOW_CONTROL Code: IF 315

Description: The user has attempted to change the Character_Flow_Control connection attribute and has specified a non-boolean value.

User Action: Specify a Character_Flow_Control value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 325-- Get operation aborted because interactive condition received.

Condition Identifier: IFES\$ABORT_GET Code: IF 325

Description: An interactive condition has occurred which caused the job to lose its terminal connection while processing an input request.

User Action: After the job's terminal connection has been restored and the job has been recovered, the input request can then be reissued.

--ERROR IF 341-- Input Editing Mode {text} must be either: normal_edit or trans_edit

Condition Identifier: IFES\$UNKNOWN_INPUT_EDITING_MODE Code: IF 341

Description: The user has selected an Input_Editing_Mode which is not a valid option for this connection attribute change.

User Action: Specify an Input_Editing_Mode option of normal or transparent and reissue the connection attribute change.

--ERROR IF 345-- End line positioning {text} must be either: none, crs, lfs, or crslfs.

Condition Identifier: IFES\$UNKNOWN_END_LINE_POSITION Code: IF 345

Description: The specified end line position must be input_positioning, carriage return, line feed, or cr_if.

User Action: Specify a correct value for end line processing.

--ERROR IF 347-- Trans Forward Character string '{text}' size must be from {text} to {text} characters.

Condition Identifier: IFES\$TRANS_FWD_CHARACTER_SIZE Code: IF 347

Description: The user has attempted to change the Trans_Forward_Character connection attribute and has specified a string whose length is outside the range of values defined by IFT\$TRANS_FWD_CHAR_SIZE.

User Action: Specify a Trans_Forward_Character string whose length is within IFT\$TRANS_FWD_CHAR_SIZE and reissue the connection attribute change.

--ERROR IF 349-- Trans Character Mode {text} must be either: none, terminate, forward, .. or forward_terminate.

Condition Identifier: IFES\$UNKNOWN_TRANS_CHAR_MODE Code: IF 349

Description: The user has selected a Trans_Character_Mode which is not a valid option for this connection attribute change.

User Action: Specify an Trans_Character_Mode option of none, terminate, forward or forward_terminate and reissue the connection attribute change.

--ERROR IF 351-- Trans Length Mode {text} must be either: none, terminate, forward, .. or forward_exact.

Condition Identifier: IFES\$UNKNOWN_TRANS_LENGTH_MODE Code: IF 351

Description: The user has selected a Trans_Length_Mode which is not a valid option for this connection attribute change.

User Action: Specify a Trans_Length_Mode option of none, terminate, forward or forward_exact and reissue the connection attribute change.

--ERROR IF 352-- Trans Timeout Mode {text} must be either: none, terminate, or forward. or forward_terminate.

Condition Identifier: IF\$UNKNOWN_TRANS_TIMEOUT_MODE Code: IF 352

Description: The user has selected a Trans_Timeout_Mode which is not a valid option for this connection attribute change.

User Action: Specify a Trans_Timeout_Mode option of none, terminate, or forward and reissue the connection attribute change.

--ERROR IF 353-- Trans Protocol Mode {text} must be either: none, terminate, or forward.

Condition Identifier: IF\$UNKNOWN_TRANS_PROTOCOL_MODE Code: IF 353

Description: The user has selected a Trans_Protocol_Mode which is not a valid option.

User Action: Specify a Trans_Protocol_Mode of none, terminate, or forward and reissue the request.

--ERROR IF 356-- Trans Terminate Character string '{text}' size must be from {text} to {text} characters.

Condition Identifier: IF\$TRANS_TERM_CHARACTER_SIZE Code: IF 356

Description: The user has attempted to change the Trans_Terminate_Character connection attribute and has specified a string whose length is outside the range of values defined by IFT\$TRANS_TERM_CHAR_SIZE.

User Action: Specify a Trans_Terminate_Character string whose length is within IFT\$TRANS_TERM_CHAR_SIZE and reissue the connection attribute change.

--ERROR IF 360-- Input Timeout must be TRUE or FALSE.

Condition Identifier: IF\$INPUT_TIMEOUT Code: IF 360

Description: The user has attempted to change the Input_Timeout connection attribute and has specified a non-boolean value.

User Action: Specify an Input_Timeout value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 365-- Input Timeout Purge must be TRUE or FALSE.

Condition Identifier: IF\$INPUT_TIMEOUT_PURGE Code: IF 365

Description: The user has attempted to change the Input_Timeout_Purge connection attribute and has specified a non-boolean value.

User Action: Specify an Input_Timeout_Purge value of TRUE or FALSE and reissue the connection attribute change.

--ERROR IF 370-- Input Timeout Length value {text} must be from {text} to {text}.

Condition Identifier: IF\$INPUT_TIMEOUT_LENGTH Code: IF 370

Description: The user has attempted to change the Input_Timeout_Length connection attribute and has specified a length that is outside the range of values defined by IFT\$INPUT_TIMEOUT_LENGTH.

User Action: Specify an Input_Timeout_Length that is within IFT\$INPUT_TIMEOUT_LENGTH and reissue the connection attribute change.

--ERROR IF 371-- Attention Character Action value {text} must be from {text} to {text}.

Condition Identifier: IF\$ATTENTION_CHARACTER_ACTION Code: IF 371

Description: The user has attempted to change the Attention_Character_Action connection attribute and has specified a value that is outside the range of values defined by IFT\$ATTENTION_CHARACTER_ACTION.

User Action: Specify an Attention_Character_Action that is within IFT\$ATTENTION_CHARACTER_ACTION and reissue the connection attribute change.

--ERROR IF 372-- Break Key Action value {text} must be from {text} to {text}.

Condition Identifier: IFEBREAK_KEY_ACTION Code: IF 372

Description: The user has attempted to change the Break_Key_Action connection attribute and has specified a value that is outside the range of values defined by IFT\$BREAK_KEY_ACTION.

User Action: Specify a Break_Key_Action that is within IFT\$BREAK_KEY_ACTION and reissue the connection attribute change.

--ERROR IF 373-- Input Block Size value {text} must be from {text} to {text}.

Condition Identifier: IFESINPUT_BLOCK_SIZE Code: IF 373

Description: The user has attempted to change the Input_Block_Size terminal attribute and has specified a value that is outside the range of values defined by IFT\$INPUT_BLOCK_SIZE.

User Action: Specify an Input_Block_Size value that is within IFT\$INPUT_BLOCK_SIZE and reissue the terminal attribute change.

--ERROR IF 390-- Wait for input from terminal exceeded Input Timeout Length.

Condition Identifier: IFESINPUT_TIMEOUT_EXCEEDED Code: IF 390

Description: This indicates that the INPUT_TIMEOUT_LENGTH attribute value has expired without terminal input being received in response to an interactive get.

User Action: Check INPUT_TIMEOUT_LENGTH and enter input before timeout length expires.

--ERROR IF 395-- Input Timeout Length is zero and there is no input available from the terminal.

Condition Identifier: IFESNO_DATA_AVAILABLE Code: IF 395

Description: This indicates that input has been requested on a file with an input timeout length of 0, and no data was available at the time of the request.

User Action: Insure that input is available when application requests it. Type ahead to do this.

--ERROR IF 400-- Input is currently expected to be in transparent mode but non-transparent input has been received.

Condition Identifier: IFESXPT_MODE_DROP_UNEXPECTED Code: IF 400

Description: The header for the last input message from the terminal indicates that the message was not delivered in transparent mode although NOS/VE Interactive Facility expected it to be. This can occur if type ahead input occurred before the Interactive Facility placed the terminal in transparent mode.

User Action: Physically break the connection by hanging up the phone, turning off the terminal, etc. Then log in a new job and attach the disconnected job. If this does not clear the condition, report the problem to an analyst. Also, make sure that the input causing the error is not entered in typed ahead fashion.

--ERROR IF 490-- The Working_Storage_Length parameter is outside the range defined by AMT\$WORKING_STORAGE_LENGTH.

Condition Identifier: IFESWSL_OUT_OF_RANGE Code: IF 490

Description: AMP\$PUT_NEXT, AMP\$PUT_PARTIAL, or AMP\$PUT_DIRECT has been called for an interactive terminal file with the working_storage_length parameter outside the range defined by AMT\$WORKING_STORAGE_LENGTH.

User Action: Respecify the working_storage_length parameter and execute the AMP\$ call again.

--ERROR IF 500-- Request for terminal characteristics failed.

Condition Identifier: IFESREQUEST_TERM_CHAR_FAIL Code: IF 500

Description: Either an attempt to LOGIN to dual state NOS/VE or an attempt to change a Terminal_Class attribute has aborted; the request for the terminal's default terminal attributes failed.

User Action: Report the problem to an analyst along with the job log if attainable.

--ERROR IF 505-- The Working_Storage_Area pointer is NIL.

Condition Identifier: IFESAIS_IS_NIL Code: IF 505

Description: AMP\$PUT_NEXT, AMP\$PUT_PARTIAL, or AMP\$PUT_DIRECT has been called with the working_storage_length parameter set to NIL.

User Action: Respecify the working_storage_length parameter and execute the AMP\$ call again.

--ERROR IF 506-- The Connection was broken before the job monitor task started.

Condition Identifier: IFESABORT_JOB_INITIALIZATION Code: IF 506

Description: The connection to the interactive terminal has been broken during login and before the job monitor task has started. The job monitor task will abort and the connection will be ended, thus making the job nonexistent.

User Action: Not applicable.

--INFORMATIVE IF 511-- A pause break condition has occurred.

Condition Identifier: IFESPAUSE_BREAK Code: IF 511

Description: A pause break condition has occurred.

User Action: None.

--INFORMATIVE IF 512-- A terminate break condition has occurred.

Condition Identifier: IFESTERMINATE_BREAK Code: IF 512

Description: A terminate break condition has occurred.

User Action: None.

--INFORMATIVE IF 513-- A terminal connection broken condition has occurred.

Condition Identifier: IFESTERMINAL_CONNECTION_BROKEN Code: IF 513

Description: A terminal connection broken condition has occurred.

User Action: None.

--INFORMATIVE IF 514-- A job reconnect to terminal condition has occurred.

Condition Identifier: IFESJOB_RECONNECT Code: IF 514

Description: A job has reconnected to a terminal.

User Action: None.

--ERROR IP 1-- Invalid port number specified in PORT command.

Condition Identifier: IPE\$BAD_PORT_COMMAND Code: IP 1

Description: The remote host has sent an invalid port number in the PORT command.

User Action: Contact site administrator.

--ERROR IP 2-- Connection with host {text} must be closed before opening a new connection.

Condition Identifier: IPE\$CMD_CONNECTION_ACTIVE Code: IP 2

Description: The existing connection with a remote host must be terminated before a new connection with another remote host can be established.

User Action: Close the existing connection with the CRECFC Utility subcommand DELETE_HOST_CONNECTION before retrying the CRECFC Utility subcommand CREATE_HOST_CONNECTION to open a new connection.

--ERROR IP 3-- The FTP command from host {text} exceeds the size limit (4096 bytes).

Condition Identifier: IPE\$COMMAND_SIZE_LIMIT_EXCEEDED Code: IP 3

Description: The FTP command received from the specified host exceeds the maximum size permitted by the FTP protocol.

User Action: Contact site administrator.

--ERROR IP 4-- Host {text} reported a connection error: {text} ({text}).

Condition Identifier: IPE\$CONNECTION_ERROR Code: IP 4

Description: The remote server host reported a connection error as indicated by the error text.

User Action: Take appropriate corrective action and retry.

--ERROR IP 5-- Connection with host {text} is terminated: {text}.

Condition Identifier: IPE\$CONTROL_CONNECTION_ERROR Code: IP 5

Description: The connection with the remote server host is terminated for the indicated reason.

User Action: Reconnect to the remote host and retry.

--ERROR IP 6-- A connection must be established prior to using the {text} command.

Condition Identifier: IPE\$CONNECTION_REQUIRED_FOR_CMD Code: IP 6

Description: A connection to a remote server host must be established prior to using the indicated CRECFC utility subcommand.

User Action: Create a connection to the destination server host and retry.

--ERROR IP 7-- Data transfer with host {text} is terminated: {text}.

Condition Identifier: IPE\$DATA_CONNECTION_ERROR Code: IP 7

Description: Data transfer with the remote server host is aborted for the indicated reason.

User Action: Take appropriate corrective action and retry.

--ERROR IP 8-- File access error on file {text}.

Condition Identifier: IPE\$FILE_ACCESS_ERROR Code: IP 8

Description: The specified file could not be accessed.

User Action: Ensure file permissions are correct. If error persists, contact site administrator.

--ERROR IP 9-- {text}

Condition Identifier: IPE\$FTP_ERROR Code: IP 9

Description: Self-explanatory.

User Action: Self-explanatory.

--CATASTROPHIC IP 10-- Internal logic error in {text}.

Condition Identifier: IPE\$FTP_INTERNAL_ERROR Code: IP 10

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 11-- File {text} is not a mass storage file and therefore cannot be used to receive binary data.

Condition Identifier: IPE\$IMAGE_XFER_REQUIRES_MS_FILE Code: IP 11

Description: Self-explanatory.

User Action: Take appropriate corrective action and retry.

--ERROR IP 13-- Received an invalid reply from host {text}: '{text}' ({text}).

Condition Identifier: IPE\$INVALID_REPLY_CODE Code: IP 13

Description: The remote FTP server has violated the FTP protocol by sending an unrecognizable FTP reply code.

User Action: Inform site analyst.

--ERROR IP 14-- Host {text} reported a login error: {text} ({text}).

Condition Identifier: IPE\$LOGIN_ERROR Code: IP 14

Description: The remote server host has rejected the user login for the indicated reason.

User Action: Take appropriate corrective action and retry.

--CATASTROPHIC IP 15-- Insufficient memory for {text}.

Condition Identifier: IPE\$MEMORY_ALLOCATE_ERROR Code: IP 15

Description: The indicated procedure cannot acquire sufficient memory to continue processing.

User Action: Retry. If condition persists, inform a site analyst.

--INFORMATIVE IP 16-- Not connected to any host.

Condition Identifier: IPE\$NO_HOST_CONNECTION Code: IP 16

Description: Self-explanatory.

User Action: None.

--ERROR IP 17-- If you supply the {text} parameter it must contain the same number of elements as the {text} parameter.

Condition Identifier: IPE\$NON_MATCHING_SET_COUNTS Code: IP 17

Description: Self-explanatory.

User Action: Retry with the correct command.

--ERROR IP 18-- Host {text} reported a file system problem for file "{text}": {text} ({text}).

Condition Identifier: IPE\$REMOTE_FILE_SYSTEM_ERROR Code: IP 18

Description: The remote server host has reported the indicated status of the server file system vis-a-vis the requested transfer or other file system action.

User Action: Take appropriate corrective action and retry.

--ERROR IP 19-- Host {text} reported remote file "{text}" is unavailable: {text} ({text}).

Condition Identifier: IPE\$REMOTE_FILE_UNAVAILABLE Code: IP 19

Description: The remote server host has reported that the specified file is not available for the indicated reason.

User Action: Take appropriate corrective action and retry.

--ERROR IP 20-- The FTP command reply from host {text} exceeds the line limit (256 lines).

Condition Identifier: IPE\$REPLY_LINE_LIMIT_EXCEEDED Code: IP 20

Description: The multi-line FTP protocol reply message from the remote server host has exceeded the maximum number of lines allowed by FTP/VE.

User Action: Retry. If condition persists, inform a site analyst.

--ERROR IP 21-- The FTP command reply from host {text} exceeds the size limit (4096 bytes).

Condition Identifier: IPE\$REPLY_SIZE_LIMIT_EXCEEDED Code: IP 21

Description: The remote server host has sent an FTP protocol reply message that exceeds the maximum size allowed by FTP/VE.

User Action: Inform a site analyst.

--ERROR IP 22-- Host {text} reported a storage allocation problem for file "{text}": {text} ({text}).

Condition Identifier: IPE\$RMT_STORAGE_ALLOC_EXCEEDED Code: IP 22

Description: The remote server host has aborted the requested action due to insufficient storage in system.

User Action: Take appropriate corrective action on the remote server system and retry later.

--ERROR IP 23-- The file limit (100,000,000 bytes) is reached for file {text}.

Condition Identifier: IPE\$SEGMENT_LIMIT_EXCEEDED Code: IP 23

Description: The indicated file from the remote server host has exceeded the storage allocation limit allowed by FTP/VE.

User Action: None.

--ERROR IP 24-- You must have the same number of elements for parameter {text} as you have for parameter {text}.

Condition Identifier: IPE\$ST_COUNT_MUST_ALWAYS_MATCH Code: IP 24

Description: A mismatch has occurred between the specified parameters.

User Action: Retry with the correct number of parameter elements.

--ERROR IP 25-- Host {text} reported a syntax error: {text} ({text}).

Condition Identifier: IPE\$SYNTAX_ERROR Code: IP 25

Description: The remote server host has rejected the indicated FTP protocol command for the indicated reason.

User Action: Consult the remote help display from the server host for the status on the rejected command.

--INFORMATIVE IP 26-- Account ignored, not required by host {text}.

Condition Identifier: IPE\$ACCOUNT_IGNORED Code: IP 26

Description: The account number specified during the login process is not required by the remote host and has been ignored.

User Action: None.

--ERROR IP 27-- Host {text} reported an unspecified error: {text} ({text}).

Condition Identifier: IPE\$UNSPECIFIED_ERROR Code: IP 27

Description: The remote host specified reported an error.

User Action: Retry previous operation. If condition persists, contact site administrator.

--ERROR IP 202-- The given connection has already been bound to an application and port.

Condition Identifier: IPE\$IPAM_BIND_DONE Code: IP 202

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 203-- A connection has already been established.

Condition Identifier: IPE\$IPAM_CON_ESTABLISHED Code: IP 203

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 205-- A listen has been done: only an accept call is allowed.

Condition Identifier: IPE\$IPAM_LISTEN_DONE Code: IP 205

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 206-- A connection has not been established yet, no listen or connect done.

Condition Identifier: IPE\$IPAM_NO_CONNECTION Code: IP 206

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 207-- All available connection identifiers have been assigned.

Condition Identifier: IPE\$IPAM_NO_SOCKET_AVAILABLE Code: IP 207

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 208-- This connection has been closed by the network.

Condition Identifier: IPE\$IPAM_SOCKET_CLOSED Code: IP 208

Description: The connection has been terminated due to a network error.

User Action: Retry later. If error persists, contact site analyst.

--ERROR IP 209-- This connection was abnormally terminated by the network.

Condition Identifier: IPE\$IPAM_SOCKET_DISCONNECTED Code: IP 209

Description: The connection has been terminated due to a network error.

User Action: Retry. If condition persists, contact site administrator.

--ERROR IP 210-- The connection was aborted, waiting for application to issue a close.

Condition Identifier: IPE\$IPAM_WAITING_FOR_CLOSE Code: IP 210

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 211-- No application name or port has been bound to the given connection.

Condition Identifier: IPE\$IPAM_UNBOUND_SOCKET Code: IP 211

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 212-- The job identifier specified is unknown.

Condition Identifier: IPE\$IPAM_UNKNOWN_JOB Code: IP 212

Description: The job identifier specified for a connection switch attempt was not recognized.

User Action: Check job identifier in connection switch logic or contact site administrator.

--ERROR IP 213-- Internal software protocol error -- connection aborted.

Condition Identifier: IPE\$IPAM_PROTOCOL_ERROR Code: IP 213

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 214-- Sufficient memory not available.

Condition Identifier: IPE\$IPAM_INSUFFICIENT_MEMORY Code: IP 214

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 215-- The network is not available at this time.

Condition Identifier: IPE\$IPAM_NETWORK_UNAVAILABLE Code: IP 215

Description: A connection could not be established due to network unavailability or a CDCNET configuration problem.

User Action: Retry later. If problem persists, contact site analyst.

--ERROR IP 216-- The specified IP address is invalid.

Condition Identifier: IPE\$IPAM_INVALID_IP_ADDRESS Code: IP 216

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 217-- The specified connection has been rejected by the foreign host or the network.

Condition Identifier: IPE\$IPAM_SOCKET_REJECTED Code: IP 217

Description: The connection attempt has been rejected.

User Action: Retry connection later -- if problem persists, contact site analyst.

--ERROR IP 218-- The offer to switch the specified connection to another job failed.

Condition Identifier: IPE\$IPAM_SOCKET_OFFER_FAILED Code: IP 218

Description: The connection switch to another job has failed.

User Action: Check connection switch logic in parent and child jobs or contact site administrator.

--ERROR IP 219-- The operation requested by the application has timed out.

Condition Identifier: IPE\$IPAM_OP_TIMED_OUT Code: IP 219

Description: The last requested operation has been timed out due to an application or network error.

User Action: Retry later. If problem persists, contact site analyst.

--ERROR IP 220-- Unable to set timeout value for the specified socket_id.

Condition Identifier: IPE\$IPAM_CANNOT_SET_TIMEOUT Code: IP 220

Description: An internal software error has occurred.

User Action: Provide site analyst with appropriate documentation or submit a PSR.

--ERROR IP 221-- The application connection limit has been exceeded.

Condition Identifier: IPE\$IPAM_CON_LIMIT_REACHED Code: IP 221

Description: The NAM/VE application connection limit has been exceeded. No more users are allowed for this application.

User Action: Retry later. If error persists, contact site analyst.

--ERROR IP 300-- No official host name for the VE host was found in the hosts file.

Condition Identifier: IPE\$IPAM_NAME_NOT_FOUND Code: IP 300

Description: The file \$SYSTEM.TCP_IP.HOSTS contains an incorrect entry.

User Action: Contact site analyst.

--ERROR IP 301-- The host name specified cannot be found in the hosts file.

Condition Identifier: IPE\$IPAM_UNKNOWN_HOST_NAME Code: IP 301

Description: The destination server host name specified on the CREATE_HOST_CONNECTION command cannot be found in file \$SYSTEM.TCP_IP.HOSTS.

User Action: Retry the command with the correct host name. If the host name entered is known to be valid, contact the site analyst regarding configuration of the \$SYSTEM.TCP_IP.HOSTS file.

--ERROR IP 302-- The host address specified cannot be found in the hosts file.

Condition Identifier: IPE\$IPAM_UNKNOWN_HOST_ADDRESS Code: IP 302

Description: The file \$SYSTEM.TCP_IP.HOSTS contains an incorrect entry.

User Action: Contact site analyst.

--ERROR IP 303-- An ip address with an invalid format was found in the hosts file.

Condition Identifier: IPE\$IPAM_INVALID_IP_FORMAT Code: IP 303

Description: The file \$SYSTEM.TCP_IP.HOSTS contains an incorrect entry.

User Action: Contact site analyst.

--ERROR IP 304-- The service name specified cannot be found in the services file.

Condition Identifier: IPE\$IPAM_UNKNOWN_SERVICE_NAME Code: IP 304

Description: The service name represented by the application cannot be found in file \$SYSTEM.TCP_IP.SERVICES.

User Action: Contact site analyst.

--ERROR IP 305-- The hosts file contains an entry with an invalid format.

Condition Identifier: IPE\$IPAM_INVALID_HOST_ENTRY Code: IP 305

Description: The file \$SYSTEM.TCP_IP.HOSTS contains an incorrect entry.

User Action: Contact site analyst.

--ERROR IP 306-- The services file contains an entry with an invalid format.

Condition Identifier: IPE\$IPAM_INVALID_SERVICE_ENTRY Code: IP 306

Description: The file \$SYSTEM.TCP_IP.SERVICES contains an incorrect entry.

User Action: Contact site analyst.

--ERROR JM 5-- {text} command is illegal for System Job.

Condition Identifier: JME\$ILLEGAL_SYSTEM_JOB_COMMAND Code: JM 5

Description: The specified command should not be issued within the system job environment (for example, the LOGOUT command).

User Action: Do not use the specified command because it will not be executed.

--INFORMATIVE JM 10-- Job terminating normally.

Condition Identifier: JME\$JOB_TERMINATING_NORMALLY Code: JM 10

Description: The job is terminating in a normal fashion with no abnormal conditions.

User Action: None.

--INFORMATIVE JM 15-- JOB_END called unexpectedly.

Condition Identifier: JME\$JOB_END_CALLED_UNEXPECTEDLY Code: JM 15

Description: The termination phase of the job was called in a task other than the job monitor task. The termination is not executed.

User Action: None.

--INFORMATIVE JM 25-- The job was not initiated before its latest time.

Condition Identifier: JME\$JOB_INITIATED_TOO_LATE Code: JM 25

Description: The job was not initiated before its latest_run_time. As a result, the job's command file was not executed.

User Action: None.

--ERROR JM 30-- Job terminated via a TERMINATE_JOB command.

Condition Identifier: JME\$JOB_TERMINATED_VIA_COMMAND Code: JM 30

Description: The job was terminated via a TERMINATE_JOB command.

User Action: None.

--INFORMATIVE JM 40-- Following error in {text} prolog:

Condition Identifier: JME\$LOGIN_ERROR_IN_PROLOG Code: JM 40

Description: An error occurred while the system was executing a prolog during login. The text field contains the name of the prolog (i.e., USER or ACCOUNT).

User Action: Correct the error in the prolog if it is the USER prolog. Contact the administrator of the prolog in error if it is a prolog other than the USER prolog.

--INFORMATIVE JM 45-- Login aborted because of following error in {text} prolog:

Condition Identifier: JME\$LOGIN_ABORT_IN_PROLOG Code: JM 45

Description: An error occurred while the system was executing a prolog during login. The login attempt was aborted because the job was batch mode or the error occurred in the SYSTEM or CLASS prolog. The text field contains the name of the prolog.

User Action: Correct the error in the prolog if it is the USER prolog. Contact the administrator of the prolog in error if it is a prolog other than the USER prolog.

--ERROR JM 1010-- The job does not exist.

Condition Identifier: JME\$NON_EXISTENT_JOB Code: JM 1010

Description: The job specified on an operator SWAP_OUT_JOB command does not exist.

User Action: Check for the correct job name of the job to be swapped out.

--ERROR JM 1020-- The job being swapped in is not in swap list.

Condition Identifier: JME\$JOB_NOT_IN_SWAP_LIST Code: JM 1020

Description: The job specified on an operator SWAP_IN_JOB command no longer exists.

User Action: None.

--ERROR JM 1030-- The job cannot be swapped out now.

Condition Identifier: JME\$JOB_CANT_BE_SWAPPED Code: JM 1030

Description: The job specified on the operator SWAP_OUT_JOB command is currently in a non-swappable state. A possible reason for the job being non-swappable is that the job is in a phase of initiation or termination and does not have a swap file allocated.

User Action: Try the swapout at a later time. Jobs that stay for a long time in a non-swappable state during initiation indicate that the system is low on disk space for swap files.

--ERROR JM 1031-- The swap buffer is full. Try again.

Condition Identifier: JME\$SWAP_BUFFER_FULL Code: JM 1031

Description: The operator SWAP_OUT_JOB/SWAP_IN_JOB cannot be handled at this time because the swap buffer is full with other jobs awaiting operator swapout/swapin.

User Action: Try the SWAP_OUT_JOB/SWAP_IN_JOB at a later time.

--ERROR JM 1032-- Job is in memory or is being swapped in.

Condition Identifier: JME\$JOB_IN_MEMORY_OR_SWAPIN Code: JM 1032

Description: The job specified on the operator SWAP_IN_JOB command is already in memory or swapping in.

User Action: None.

--ERROR JM 1033-- Job has ijl status of non-swappable.

Condition Identifier: JME\$JOB_STATUS_NON_SWAPPABLE Code: JM 1033

Description: The job specified on the operator SWAP_OUT_JOB command is currently in a non-swappable state. The job is in a phase of initiation or termination and does not have a swap file allocated.

User Action: Try the swapout at a later time. Jobs that stay for a long time in a non-swappable state during initiation indicate that the system is low on disk space for swap files.

--ERROR JM 1043-- Swapping has been disabled.

Condition Identifier: JME\$SWAPPING_NOT_ALLOWED Code: JM 1043

Description: The job specified on the operator SWAP_IN_JOB or SWAP_OUT_JOB command cannot be swapped.

User Action: Don't try to swap the job.

--ERROR JM 1045-- Job is dead because of io errors, cannot swap.

Condition Identifier: JME\$JOB_DEAD_CANNOT_SWAP Code: JM 1045

Description: I/O errors occurred while reading or writing the job's swap file. The job can no longer be swapped.

User Action: None.

--ERROR JM 1056-- The specified dispatching priority is invalid.

Condition Identifier: JME\$INVALID_DISPATCH_PRIORITY Code: JM 1056

Description: The dispatching priority specified on the operator CHANGE_PRIORITY or user CHANGE_JOB_ATTRIBUTE command is invalid.

User Action: Specify a valid dispatching priority on the command.

--WARNING JM 1058-- ATTENTION OPERATOR: The number of maximum active jobs for the service class {text} for the job {text} is currently set to zero. It must be changed before the swapin can proceed.

Condition Identifier: JME\$SWAPIN_WITH_MAXAJ_ZERO Code: JM 1058

Description: The operator is trying to swap a job into memory using the SWAP_IN_JOB command, but the MAXIMUM_ACTIVE_JOBS attribute for the service class is zero.

User Action: Set the MAXIMUM_ACTIVE_JOBS attribute for the service class to a value greater than zero.

--ERROR JM 1065-- The access identification given does not match the one using the utility.

Condition Identifier: JME\$ACCESS_ID_MISMATCH Code: JM 1065

Description: A request to install or update the scheduling profile in the scheduler tables is being validated. The access identification provided by the MANAGE_ACTIVE_SCHEDULING utility is incorrect and access to the scheduler tables is not permitted. The scheduler tables remain unchanged. This error may occur when the processing of the request is interrupted by a system termination and then recovered during a continuation deadstart.

User Action: Retry the request.

--ERROR JM 1066-- Another scheduling utility is currently active.

Condition Identifier: JME\$ANOTHER_UTILITY_IS_ACTIVE Code: JM 1066

Description: The MANAGE_ACTIVE_SCHEDULING command cannot be processed because another copy of this utility is currently active. Only one copy of this utility may be in use at a given time.

User Action: Retry the request later.

--ERROR JM 1067-- The application names are not sorted in ascending order.

Condition Identifier: JME\$APPLICATIONS_NOT_SORTED Code: JM 1067

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The set of applications to be installed is not sorted in ascending order as required. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1068-- The {text} abbreviation {text} is not unique.

Condition Identifier: JME\$CLASS_ABBREV_NOT_UNIQUE Code: JM 1068

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated abbreviation for a job class or service class is not unique among all other names and abbreviations of its class kind. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Review and correct the scheduling profile that is being activated and retry the request.

--ERROR JM 1069-- The {text} {text} is already defined with class index {text}.

Condition Identifier: JME\$CLASS_INDEX_ALREADY_IN_USE Code: JM 1069

Description: A request to install a scheduling profile in the system scheduler tables is being validated. A job class or service class to be installed has a class index that is already assigned to the class indicated in the message. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1070-- The {text} {text} to be updated has an incorrect class index.

Condition Identifier: JME\$CLASS_INDEX_CONFLICT Code: JM 1070

Description: A request to install or update a scheduling profile in the system scheduler tables is being validated. The indicated job class or service class in the new profile is defined in the scheduler tables with a different class index. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1071-- The {text} index {text} is not defined.

Condition Identifier: JME\$CLASS_INDEX_NOT_DEFINED Code: JM 1071

Description: A request to access or modify the indicated job class, service class or application in the system scheduler tables is rejected because the class or application index provided is not defined in the scheduler tables. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1072-- The {text} {text} is not defined.

Condition Identifier: JME\$CLASS_OR_APPL_NOT_DEFINED Code: JM 1072

Description: A request to update the scheduling profile in the system scheduler tables is being validated. The indicated job class, service class, or application to be updated is not defined in the scheduler tables. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1073-- The {text} name {text} is not unique.

Condition Identifier: JME\$CLASS_OR_APPL_NOT_UNIQUE Code: JM 1073

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated name for a job class, service class, or application is not unique for its class or application kind. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Review and correct the scheduling profile that is being activated and retry the request.

--ERROR JM 1074-- The scheduling profile cannot be activated because the {text} {text} being deleted has active jobs.

Condition Identifier: JME\$DELETE_CLASS_STILL_ACTIVE Code: JM 1074

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated job class or service class is not defined in the new profile and is to be deleted from the scheduler tables. It cannot be deleted until there are no active jobs in it. The scheduler tables remain unchanged.

User Action: Retry the request when all jobs have completed execution in the indicated class. If appropriate to site policy, use the `MANAGE_ACTIVE_SCHEDULING` utility to disable the initiation of new jobs for the indicated job class or for all job classes using the indicated service class.

--ERROR JM 1075-- The ranking of the job classes is incorrect.

Condition Identifier: JME\$ERROR_IN_JOB_CLASS_RANKING Code: JM 1075

Description: A request for information about the job classes defined in the system scheduler tables cannot be processed because the ranking of the job classes in the job class table is indeterminate. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software or job class table. Refresh the system scheduler tables by activating the active scheduling profile under the `MANAGE_ACTIVE_SCHEDULING` utility with the command: `ACTIVATE_PROFILE $system.scheduling.osf$system_profile.2`

--ERROR JM 1076-- The {text} {text} in the scheduler table is not in the profile being installed.

Condition Identifier: JME\$EXCESS_CLASS_IN_SCHED_TABLE Code: JM 1076

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated job class or service class is an excess class in the scheduler tables because it is not defined in the new profile and is not marked for deletion. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1077-- The job class specified is not defined.

Condition Identifier: JME\$JOB_CLASS_NOT_DEFINED Code: JM 1077

Description: A request to determine if a specified job class name or index is defined in the system scheduler tables was processed. The job class is not defined.

User Action: None.

--ERROR JM 1078-- Scheduling administration privilege is required to use this request.

Condition Identifier: JME\$MUST_BE_SCHEDULING_ADMIN Code: JM 1078

Description: Scheduling administration privilege is required to use the request given. Your user name is not validated for this privilege.

User Action: If you have a requirement for scheduling administration privilege, contact your validation administrator.

--ERROR JM 1080-- The predefined {text} {text} cannot be deleted.

Condition Identifier: JME\$NO_DELETE_OF_DEFAULT_CLASS Code: JM 1080

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated job class or service class is not defined in the new profile and is to be deleted from the scheduler tables. This class is a default class established during system deadstart and deletion of it is not permitted. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Review and correct the scheduling profile that is being activated and retry the request.

--ERROR JM 1081-- Unable to access an element in a sequence.

Condition Identifier: JME\$NO_ELEMENT_IN_SEQUENCE Code: JM 1081

Description: A CYBIL SEQUENCE storage structure is expected to contain an element and it does not. If a request to install or update a scheduling profile in the system scheduler tables was being processed, the scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1082-- The predefined {text} {text} cannot be used for automatic class selection.

Condition Identifier: JME\$NO_RANKING_OF_DEFAULT_CLASS Code: JM 1082

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated job class cannot be designated for automatic class selection during job submission because it is a default job class defined during system deadstart. Job submission to these job classes must be requested explicitly by job class name. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Review and correct the scheduling profile being activated and retry the request.

--ERROR JM 1083-- Unable to allocate a variable in the run-time stack.

Condition Identifier: JME\$NO_SPACE_IN_RUNTIME_STACK Code: JM 1083

Description: A CYBIL PUSH statement did not successfully allocate space for a variable in the run-time stack. If a request to install or update a scheduling profile in the system scheduler tables was being processed, the scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1084-- The utility active flag must be set in order to change the scheduler tables.

Condition Identifier: JME\$NO_UTILITY_IS_ACTIVE Code: JM 1084

Description: A request to install or update the scheduling profile in the system scheduler tables is being validated. The sequence of operations executed by the MANAGE_ACTIVE_SCHEDULING utility is incorrect and access to the scheduler tables is not permitted. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software.

--ERROR JM 1085-- The scheduling profile cannot be read.

Condition Identifier: JME\$PROFILE_CANNOT_BE_READ Code: JM 1085

Description: This message is issued to the SYSTEM log during system deadstart when the active scheduling profile permanent file cannot be read. Normally, the maximum job class and service class indices from this file are used to allocate space for the scheduler tables. When the file cannot be read, table space is allocated using the maximum job classes and service classes declared by the site in the system core commands. Recovery of active jobs is disabled since the class definitions for active jobs are unknown. A WARNING message is issued to the operator console stating that the profile file cannot be accessed and that job recovery is not possible.

User Action: None is possible at this time. Later in deadstart, the MANAGE_ACTIVE_SCHEDULING utility will install the active scheduling profile in the system scheduler tables. If this attempt to read the profile also fails, it may be necessary to restore or recreate the active scheduling profile permanent file being used by the site after system deadstart is complete.

--ERROR JM 1086-- The scheduling profile was not installed and cycle 2 of the system profile has already been deleted. Retry the operation.

Condition Identifier: JME\$PROFILE_CYCLE2_LOST Code: JM 1086

Description: A request to install a scheduling profile in the system scheduler tables has been validated successfully but the scheduler tables were not changed. The previous version of the active scheduling profile permanent file has already been deleted. This error may occur when the processing of the request is interrupted by a system termination and then recovered during a continuation deadstart.

User Action: The request must be retried in order to establish the active scheduling profile permanent file.

--ERROR JM 1087-- The profile changes to be made require activation of the profile.

Condition Identifier: JME\$PROFILE_ID_MISMATCH Code: JM 1087

Description: A request to update the scheduling profile in the system scheduler tables is being validated. The nature of the changes to be made require that a new scheduling profile be prepared and activated. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Using the ADMINISTER_SCHEDULING utility, prepare a new scheduling profile with the desired changes. Install this profile in the system scheduler tables using the ACTIVATE_PROFILE subcommand of the MANAGE_ACTIVE_SCHEDULING utility. Refer to the NOS/VE System Performance and Maintenance manual, Volume 1, for a description of these utilities.

--ERROR JM 1088-- The scheduling profile was not installed. Retry the operation.

Condition Identifier: JME\$PROFILE_NOT_INSTALLED Code: JM 1088

Description: A request to install a scheduling profile in the system scheduler tables has been validated successfully, but the scheduler tables were not changed. This error may occur when the processing of the request is interrupted by a system termination and then recovered during a continuation deadstart.

User Action: Retry the request.

--ERROR JM 1089-- Maximum {text} {text} required at deadstart to install the scheduling profile.

Condition Identifier: JME\$PROFILE_TOO_LARGE Code: JM 1089

Description: The maximum number of job classes or service classes declared during system deadstart is too small to install the active scheduling profile in the system scheduler tables.

User Action: Deadstart and declare the maximum number of job classes or service classes as indicated by the error message using the SET_SYSTEM_ATTRIBUTE command. Refer to the NOS/VE System Performance and Maintenance manual, Volume 1, for a description of this command.

--ERROR JM 1092-- The service class specified is not defined.

Condition Identifier: JME\$SERVICE_CLASS_NOT_DEFINED Code: JM 1092

Description: A request to determine if a specified service class name or index is defined in the system scheduler tables was processed. The service class is not defined.

User Action: None.

--INFORMATIVE JM 1093-- In the system scheduling profile, the maximum Job Class index is {text} and the maximum Service Class index is {text}.

Condition Identifier: JME\$TABLE_LENGTHS_FROM_PROFILE Code: JM 1093

Description: This is an informative message issued to the SYSTEM log during system deadstart which gives the maximum job class and service class indices defined in the active scheduling profile permanent file. These indices may be larger than the maximum number of classes declared by the site during system deadstart. In this case, the system software automatically allocates sufficient space in the scheduler tables to accommodate the maximum class indices.

User Action: If the maximum index for job classes or service classes becomes larger than the maximum declared during system deadstart, it is advisable to recreate the active scheduling profile permanent file. Refer to the NOS/VE System Performance and Maintenance manual, Volume 1, for additional information under the ACTIVATE_PROFILE subcommand of the MANAGE_ACTIVE_SCHEDULING utility.

--ERROR JM 1094-- An unknown class kind was specified when requesting the defined class list.

Condition Identifier: JME\$UNKNOWN_CLASS_KIND Code: JM 1094

Description: A request for information about the classes defined in the system scheduler tables was unrecognizable.

User Action: Correct the request and retry.

--WARNING JM 1095-- No operation was performed. This command has been replaced by the ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING Utilities.

Condition Identifier: JME\$USE_ADMS_OR_MANAS_UTILITY Code: JM 1095

Description: No change to the system scheduler tables has been made. The operator command issued to modify the system scheduler tables has been replaced by different commands under the ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING utilities.

User Action: Refer to the NOS/VE System Performance and Maintenance manual, Volume 1, for a description of the job scheduling utilities.

--WARNING JM 1096-- No operation was performed. This command has been replaced by the ADMINISTER_SCHEDULING Utility.

Condition Identifier: JME\$USE_ADMS_UTILITY Code: JM 1096

Description: No change to the system scheduler tables has been made. The operator command issued to modify the system scheduler tables has been replaced by different commands under the ADMINISTER_SCHEDULING utility.

User Action: Refer to the NOS/VE System Performance and Maintenance manual, Volume 1, for a description of the job scheduling utilities.

--INFORMATIVE JM 2005-- {text} warning limit has been reached.

Condition Identifier: JME\$RESOURCE_CONDITION Code: JM 2005

Description: The value of the accumulator for the specified limit is greater than or equal to the job resource limit.

User Action: Raise the job resource limit or terminate the job.

--INFORMATIVE JM 2010-- Job Time Limit reached.

Condition Identifier: JME\$TIME_LIMIT_CONDITION Code: JM 2010

Description: The time limit that was set (via the SET_JOB_LIMITS command) for the job has been reached.

User Action: The user can either extend the time limit further (by specifying an additional time increment) or terminate the job (LOGOUT command).

--INFORMATIVE JM 2015-- Maximum limit value is {text}.

Condition Identifier: JME\$MAXIMUM_LIMIT_IS Code: JM 2015

Description: This message displays the current maximum for the specified limit whenever an interactive resource condition occurs.

User Action: None.

--INFORMATIVE JM 2020-- Accumulator value is {text}.

Condition Identifier: JME\$CURRENT_ACCUMULATOR_IS Code: JM 2020

Description: This message displays the current accumulator for the specified limit, whenever an interactive resource condition occurs.

User Action: None.

--INFORMATIVE JM 2040-- Please enter an integer increment or TERMINATE_COMMAND or LOGOUT.

Condition Identifier: JME\$INCREMENT_PROMPT Code: JM 2040

Description: This message prompts indicating options for responding to an interactive resource condition.

User Action: Enter an integer increment or TERMINATE_COMMAND or LOGOUT command.

--INFORMATIVE JM 2045-- The integer increment must be in the range {text} to {text}.

Condition Identifier: JME\$INCREMENT_RANGE Code: JM 2045

Description: This message is displayed if the integer increment specified for the response to an interactive resource condition is out of range.

User Action: None.

--INFORMATIVE JM 2050-- Job has been reconnected to this terminal.'

Condition Identifier: JME\$JOB_RECONNECTED Code: JM 2050

Description: This message is displayed when an interactive job is reconnected after having been disconnected while processing an interactive resource condition.

User Action: None.

--INFORMATIVE JM 2055-- {text} is not a valid integer increment or TERMINATE_COMMAND or LOGOUT.

Condition Identifier: JME\$NOT_INTEGER_OR_LOGOUT Code: JM 2055

Description: This message is displayed if an incorrect response to an interactive resource condition is entered.

User Action: None.

--INFORMATIVE JM 2060-- Pause break ignored.

Condition Identifier: JME\$PAUSE_BREAK_IGNORED Code: JM 2060

Description: This message is displayed if a pause break is entered in response to an interactive resource condition.

User Action: None.

--INFORMATIVE JM 2065-- Terminate break ignored.

Condition Identifier: JME\$TERMINATE_BREAK_IGNORED Code: JM 2065

Description: This message is displayed if a terminate break is entered in response to an interactive resource condition.

User Action: None.

--ERROR JM 4010-- The job {text} was not found.

Condition Identifier: JME\$JOB_NOT_FOUND Code: JM 4010

Description: The job specified on a command could not be found.

User Action: Check the job name that was specified.

--ERROR JM 4020-- The file {text} was not found.

Condition Identifier: JME\$FILE_NOT_FOUND Code: JM 4020

Description: The output file specified on a command could not be found.

User Action: Check the output file name that was specified.

--WARNING JM 4040-- Terminating the current job ({text}) is not permitted.

Condition Identifier: JME\$TRIED_TO_SELF_DESTRUCT Code: JM 4040

Description: The requesting job attempted to terminate itself. The request is not allowed.

User Action: Do not attempt to terminate the requesting job.

--ERROR JM 4041-- The job {text} cannot be terminated.

Condition Identifier: JME\$JOB_CANNOT_BE_TERMINATED Code: JM 4041

Description: The job indicated cannot be terminated. For example, the system job cannot be terminated.

User Action: Do not attempt to terminate the job.

--ERROR JM 4050-- Required parameter {text} omitted.

Condition Identifier: JME\$MISSING_PARAMETER Code: JM 4050

Description: A required parameter was omitted on the attempted request.

User Action: Supply the required parameter.

--ERROR JM 4060-- Specifying the key {text} for the {text} parameter on the {text} request is not meaningful or it is not supported at this time.

Condition Identifier: JME\$INVALID_PARAMETER Code: JM 4060

Description: An unexpected or unsupported value was specified within a (JM) request.

User Action: Specify only the documented parameters for the request.

--ERROR JM 4061-- The value specified for the key {text} for the {text} parameter on the {text} request is not meaningful or it is not supported at this time.

Condition Identifier: JME\$INVALID_PARAMETER_VALUE Code: JM 4061

Description: An unexpected or unsupported parameter value was specified within a (JM) request.

User Action: Specify only the documented values for a request.

--ERROR JM 4070-- Invalid keyword for {text}.

Condition Identifier: JME\$INVALID_KEYWORD Code: JM 4070

Description: The keyword supplied for a parameter is invalid.

User Action: Supply only valid keywords.

--ERROR JM 4080-- Special privilege is required.

Condition Identifier: JME\$SPECIAL_PRIVILEGE_REQUIRED Code: JM 4080

Description: An attempt was made to issue a command that only the site operator can issue.

User Action: Do not use the command.

--ERROR JM 4110-- The name {text} was not found.

Condition Identifier: JME\$NAME_NOT_FOUND Code: JM 4110

Description: The specified job name or print file name does not exist.

User Action: Recheck the specified name.

--ERROR JM 4120-- Multiple names of {text} were found. Use system supplied name to be unique.

Condition Identifier: JME\$DUPLICATE_NAME Code: JM 4120

Description: More than one version of the specified name exists.

User Action: Use a unique system supplied name.

--ERROR JM 4130-- You must be an operator to use this command.

Condition Identifier: JME\$MUST_BE_OPERATOR Code: JM 4130

Description: An attempt was made to issue a command that only the site operator can issue.

User Action: Do not use the command.

--ERROR JM 4140-- The job resources are temporarily unavailable.

Condition Identifier: JME\$MAXIMUM_JOBS Code: JM 4140

Description: The resources needed to define a job within the system are currently unavailable.

User Action: Resubmit the job, perhaps at a later time.

--ERROR JM 4150-- The output resources are temporarily unavailable.

Condition Identifier: JME\$MAXIMUM_OUTPUT Code: JM 4150

Description: The resources needed to define an output file within the system are currently unavailable.

User Action: Retry the print file action, perhaps at a later time.

--ERROR JM 4160-- There are no files in the output queue.

Condition Identifier: JME\$OUTPUT_QUEUE_IS_EMPTY Code: JM 4160

Description: An output application has attempted to acquire files from a NOS/VE output queue and the queue is empty.

User Action: Notify a site analyst.

--ERROR JM 4161-- There are no jobs in the input queue.

Condition Identifier: JME\$INPUT_QUEUE_IS_EMPTY Code: JM 4161

Description: An input application has attempted to acquire files from a NOS/VE input queue and the queue is empty.

User Action: Notify a site analyst.

--ERROR JM 4180-- Error attempting to write job system label.

Condition Identifier: JME\$WRITE_JOB_SYSTEM_LABEL Code: JM 4180

Description: A job's queue file label could not be written.

User Action: Notify a site analyst.

--ERROR JM 4185-- Error attempting to write output system label.

Condition Identifier: JME\$WRITE_OUTPUT_SYSTEM_LABEL Code: JM 4185

Description: An output file's queue file label could not be written.

User Action: Notify a site analyst.

--ERROR JM 4190-- Error attempting to read job system label.

Condition Identifier: JME\$READ_JOB_SYSTEM_LABEL Code: JM 4190

Description: A job's queue file label could not be read.

User Action: Notify a site analyst.

--ERROR JM 4195-- Error attempting to read output system label.

Condition Identifier: JME\$READ_OUTPUT_SYSTEM_LABEL Code: JM 4195

Description: An output file's queue file label could not be read.

User Action: Notify a site analyst.

--ERROR JM 4200-- Job {text} has already been terminated via command.

Condition Identifier: JME\$JOB_ALREADY_TERMINATED Code: JM 4200

Description: The specified job has already been terminated by a previous TERMINATE_JOB command.

User Action: None.

--WARNING JM 4201-- The job {text} has been forced out of memory and cannot be terminated.

Condition Identifier: JME\$JOB_FORCED_OUT_OF_MEMORY Code: JM 4201

Description: The job you are trying to terminate was forced out of memory and cannot be terminated.

User Action: If the operator can swap the job in with the SWAP_IN_JOB command, it can be terminated. Otherwise, there is no user action.

--WARNING JM 4202-- The job {text} has a task that cannot be terminated therefore the job cannot be terminated.

Condition Identifier: JME\$JOB_HAS_A_HUNG_TASK Code: JM 4202

Description: One of the tasks in the job is hung so that the job cannot be terminated.

User Action: None.

--ERROR JM 4210-- Output file {text} has already been terminated via command.

Condition Identifier: JME\$OUTPUT_ALREADY_TERMINATED Code: JM 4210

Description: The specified output file has already been terminated by a previous TERMINATE_OUTPUT command.

User Action: None.

--ERROR JM 4215-- The user supplied name cannot be null.

Condition Identifier: JME\$NO_USER_NAME_SPECIFIED Code: JM 4215

Description: The caller of a JM program interface procedure has specified a keyword of user_supplied_name and has not specified a value for that name.

User Action: Specify a value for the name.

--ERROR JM 4220-- The job state cannot be null.

Condition Identifier: JME\$JOB_STATE_IS_NULL Code: JM 4220

Description: The caller of a JM program interface procedure has specified a keyword of job_state and has not initialized the value to anything valid or has initialized the value to be empty.

User Action: Specify a value for job_state.

--ERROR JM 4230-- No jobs were found.

Condition Identifier: JME\$NO_JOBS_WERE_FOUND Code: JM 4230

Description: No jobs fit the selection criteria specified on the SELECT_JOBS subcommand of the MANAGE_JOBS utility.

User Action: Verify that the correct selection criteria were specified.

--ERROR JM 4240-- No outputs were found.

Condition Identifier: JME\$NO_OUTPUTS_WERE_FOUND Code: JM 4240

Description: No output files fit the selection criteria specified for the SELECT_OUTPUT subcommand of the MANAGE_OUTPUT utility.

User Action: Verify that the correct selection criteria were specified.

--ERROR JM 4270-- VERSION in System Label does not match running system.

Condition Identifier: JME\$SL_VERSION_MISMATCH Code: JM 4270

Description: The queue file label of the file does not match the value of the running system.

User Action: No action required.

--ERROR JM 4280-- The name {text} is an illegal SYSTEM SUPPLIED NAME.

Condition Identifier: JME\$ILLEGAL_SSN Code: JM 4280

Description: The caller of a JM program interface procedure has specified a name as a system_supplied_name and the name is not a valid system_supplied_name.

User Action: Specify the name correctly.

--ERROR JM 4285-- It is improper to use {text} as a user supplied name.

Condition Identifier: JME\$ILLEGAL_USN Code: JM 4285

Description: An attempt was made to specify a user_supplied_name for a job or output file that is of the same form as a system_supplied_name.

User Action: Specify a name that is not in the form of a system_supplied_name.

--ERROR JM 4290-- The key {text} was specified more than once for the {text} parameter on the {text} request.

Condition Identifier: JME\$DUPLICATE_ATTRIBUTE_KEY Code: JM 4290

Description: An attribute key was duplicated on a (JM) request that does not support duplication of key selectors.

User Action: Do not duplicate attribute key selectors on the request.

--INFORMATIVE JM 4297-- The input file {text} was recovered.

Condition Identifier: JME\$INPUT_WAS_RECOVERED Code: JM 4297

Description: The input file was recovered.

User Action: No action required.

--INFORMATIVE JM 4298-- The input file {text} was not recovered.

Condition Identifier: JME\$INPUT_WAS_NOT_RECOVERED Code: JM 4298

Description: The input file was not recovered due to an error.

User Action: No action required.

--INFORMATIVE JM 4299-- The job_recovery_disposition or the job_abort_disposition indicated that the job should be {text}.

Condition Identifier: JME\$JOB_RECOVERY_OR_ABORT_SET Code: JM 4299

Description: The job's recovery or abort disposition indicated that the job should be terminated.

User Action: No action required.

--ERROR JM 4300-- The catalog {text} for the family {text} could not be found.

Condition Identifier: JME\$UNABLE_TO_RECOVER_CATALOG Code: JM 4300

Description: The input or output catalog could not be recovered for the family indicated during deadstart.
This may be due to a damaged condition or a missing volume.

User Action: No action required.

--INFORMATIVE JM 4301-- The job {text} was recovered.

Condition Identifier: JME\$JOB_WAS_RECOVERED Code: JM 4301

Description: Job recovery was successfully completed.

User Action: No action required.

--INFORMATIVE JM 4302-- The job {text} was not recovered.

Condition Identifier: JME\$JOB_WAS_NOT_RECOVERED Code: JM 4302

Description: Job recovery was not successfully completed.

User Action: No action required.

--INFORMATIVE JM 4303-- The output file {text} was recovered.

Condition Identifier: JME\$OUTPUT_WAS_RECOVERED Code: JM 4303

Description: The output file was recovered.

User Action: No action required.

--INFORMATIVE JM 4304-- The output file {text} was not recovered.

Condition Identifier: JME\$OUTPUT_WAS_NOT_RECOVERED Code: JM 4304

Description: Due to an error, the output file was not recovered.

User Action: No action required.

--ERROR JM 4305-- Must be the system job to use {text}.

Condition Identifier: JME\$MUST_BE_SYSTEM_JOB Code: JM 4305

Description: In order to perform the request, the job must be the system job.

User Action: Only attempt the request from the system job.

--ERROR JM 4310-- Can't recover {text} job.

Condition Identifier: JME\$CANT_RECOVER_JOB Code: JM 4310

Description: The job could not be recovered for the indicated reason.

User Action: None.

--ERROR JM 4315-- {text} is not a valid job class.

Condition Identifier: JME\$INVALID_JOB_CLASS Code: JM 4315

Description: The job class specified is not a defined job class.

User Action: Use only defined job classes.

--ERROR JM 4320-- The output {text} is printing and cannot be changed.

Condition Identifier: JME\$OUTPUT_IS_INITIATED Code: JM 4320

Description: The request cannot be performed on the output file because it has already started printing or is in the process of being transferred.

User Action: Only attempt the request on files that are not printing.

--ERROR JM 4321-- The input file {text} has been initiated and cannot be changed.

Condition Identifier: JME\$INPUT_IS_INITIATED Code: JM 4321

Description: The request cannot be performed on the input file because the job is initiated or is in the process of being transferred.

User Action: Only attempt this request on jobs that have not initiated.

--ERROR JM 4322-- The input attributes of an interactive mode job cannot be changed.

Condition Identifier: JME\$CANNOT_CHANGE_INTERACTIVE Code: JM 4322

Description: The input attributes of an interactive job cannot be changed.

User Action: Do not attempt to change the attributes of an interactive job.

--ERROR JM 4325-- The output {text} is terminated and cannot be changed.

Condition Identifier: JME\$OUTPUT_IS_TERMINATED Code: JM 4325

Description: The request cannot be performed on the output file because it has been terminated.

User Action: Only attempt the request on files that have not been terminated.

--ERROR JM 4330-- The output file requested cannot be initiated and must be reacquired.

Condition Identifier: JME\$OUTPUT_CANNOT_INITIATE Code: JM 4330

Description: An application has attempted to start transfer of a file that has been modified since the application acquired it.

User Action: Notify a site analyst.

--ERROR JM 4331-- The input file requested cannot be initiated and must be reacquired.

Condition Identifier: JME\$INPUT_CANNOT_INITIATE Code: JM 4331

Description: An application has attempted to start transfer of an input file that has been modified since the application has acquired it.

User Action: Notify a site analyst.

--ERROR JM 4340-- A permanent file is required for the OUTPUT_ DISPOSITION parameter.

Condition Identifier: JME\$PERMANENT_FILE_REQUIRED Code: JM 4340

Description: A file specified for standard_output must be a permanent file.

User Action: Specify a permanent file.

--ERROR JM 4350-- \$NULL cannot be used as input to this command.

Condition Identifier: JME\$CANT_USE_\$NULL Code: JM 4350

Description: \$NULL is not a valid file name.

User Action: Specify some other file name.

--ERROR JM 4360-- The value specified for {text} is out of range.

Condition Identifier: JME\$VALUE_OUT_OF_RANGE Code: JM 4360

Description: The value specified is not in the acceptable range.

User Action: Specify values that are in the proper range.

--ERROR JM 4365-- It is not valid to have a minimum_working_set value greater than the maximum_working_set value.

Condition Identifier: JME\$INVALID_WORKING_SET_SIZE Code: JM 4365

Description: The value specified for the minimum_working_set is greater than the value for the maximum_working_set.

User Action: Do not specify a minimum_working_set greater than maximum_working_set.

--ERROR JM 4370-- A data_mode of transparent is not valid with a destination_usage of dual_state.

Condition Identifier: JME\$INVALID_DATA_MODE Code: JM 4370

Description: Transparent files are not supported by the interface to the dual_state partner.

User Action: Do not print transparent files with a destination usage of dual_state.

--ERROR JM 4380-- The requested operation cannot be performed when the job is in termination.

Condition Identifier: JME\$JOB_IS_IN_TERMINATION Code: JM 4380

Description: The operation you attempted, such as input/output to a terminal, has occurred during job termination. For example, when the terminal has been disconnected and the job is timing out, the users epilog now requests input.

User Action: The user needs to check for this status if some portion of his code, executed during task or job termination, requests interaction with the user.

--ERROR JM 4381-- Exit due to user request <NCC>-X for logout.

Condition Identifier: JME\$USER_REQUESTED_EXIT Code: JM 4381

Description: The user has requested logout by using the network_command_character (NCC) followed by an X.

User Action: None.

--ERROR JM 4382-- The job {text} cannot be attached because it originated from a non-compatible network.

Condition Identifier: JME\$INCOMPATIBLE_NETWORK_ORIGIN Code: JM 4382

Description: The job specified in the ATTACH_JOB command cannot be attached because it originated from a network with a different pathway to NOS/VE. That is, if a job was logged in via the dual-state pathway and was detached, you must login to NOS/VE via dual-state in order to attach the job.

User Action: Log in via the same network as the detached job.

--INFORMATIVE JM 4383-- Terminal timeout completes at {text}.{text}.

Condition Identifier: JME\$TERMINAL_TIMEOUT_MESSAGE Code: JM 4383

Description: Interactive timeout completes at the indicated time.

User Action: None.

--ERROR JM 4405-- The file's LATEST_PRINT_TIME has expired.

Condition Identifier: JME\$LATEST_PRINT_TIME_EXPIRED Code: JM 4405

Description: The latest print time for the file has already occurred.

User Action: Change the latest print time to a future date and time.

--ERROR JM 4500-- The specified output state is invalid.

Condition Identifier: JME\$INVALID_OUTPUT_STATE Code: JM 4500

Description: The output state specified on the request is invalid.

User Action: Only specify valid output states on the request.

--ERROR JM 4501-- The specified job state is invalid.

Condition Identifier: JME\$INVALID_JOB_STATE Code: JM 4501

Description: The job state specified on the request is invalid.

User Action: Only specify valid job states on the request.

--ERROR JM 4505-- The destination usage {text} is incorrect.

Condition Identifier: JME\$DESTINATION_USAGE_INCORRECT Code: JM 4505

Description: An application has attempted to register for an invalid DESTINATION_USAGE.

User Action: Notify a site analyst.

--ERROR JM 4510-- The requesting application is not permitted access to output files with a destination usage of {text}.

Condition Identifier: JME\$APPLICATION_NOT_PERMITTED Code: JM 4510

Description: An application (or user) has attempted to make a request that it is not permitted to perform.

User Action: Notify a site analyst.

--ERROR JM 4515-- The destination_usage {text} is already in use by another application.

Condition Identifier: JME\$DESTINATION_USAGE_IN_USE Code: JM 4515

Description: An application has attempted to register for a DESTINATION_USAGE for which another application is currently registered.

User Action: Notify a site analyst.

--ERROR JM 4520-- The maximum number of applications have already registered.

Condition Identifier: JME\$APPLICATION_TABLE_IS_FULL Code: JM 4520

Description: There is not enough room to register any additional applications.

User Action: Notify a site analyst.

--ERROR JM 4525-- The parameter {text} is required when the parameter {text} is specified.

Condition Identifier: JME\$PARAMETER_REQUIRED_WHEN Code: JM 4525

Description: A parameter was specified that requires a value to be given for another parameter.

User Action: Specify the required parameter on the command.

--ERROR JM 4530-- Interactive job discarded because it could not be reassigned to a Job Class.

Condition Identifier: JME\$INTERACTIVE_JOB_DISCARDED Code: JM 4530

Description: A new scheduling profile was installed in the system scheduler tables. There is no job class in the new profile that satisfies the requirements of an interactive job that was waiting to log into the system. Since the job cannot complete logging in until reassigned to a job class, it is terminated to avoid leaving the terminal user waiting for a response. This error message is returned to the caller installing the new profile. It is not returned to the interactive terminal user.

User Action: None.

--ERROR JM 4535-- Not all jobs in Job Class {text} were moved to the UNASSIGNED Job Class.

Condition Identifier: JME\$NOT_ALL_JOBS_WERE_MOVED Code: JM 4535

Description: A request to install a scheduling profile in the system scheduler tables is being validated. The indicated job class is not defined in the new profile and is to be deleted from the scheduler tables. There are input jobs queued in this class that cannot be moved to the UNASSIGNED job class. The new profile cannot be installed until all of the jobs queued in this class can be moved. The scheduler tables remain unchanged. This error is not normally expected to occur.

User Action: Retry the request. If the error persists, it may be a failure in the system software or queued file tables.

--ERROR JM 4540-- The scheduling profile changed while a job was being submitted.

Condition Identifier: JME\$SCHEDULING_PROFILE_CHANGED Code: JM 4540

Description: A new scheduling profile was installed in the system scheduler tables while the job was being evaluated for submission to the input queue. This error is processed by the system software and the job is automatically re-submitted when the updating of the scheduler tables is complete.

User Action: None.

--ERROR JM 4545-- User not valid for access to batch services.

Condition Identifier: JME\$BATCH_ACCESS_DENIED Code: JM 4545

Description: Your user name is not validated for batch access to the system. Therefore, jobs cannot be submitted for execution using the SUBMIT_JOB and JOB/JOBEND commands.

User Action: If you have a requirement for batch services, contact your validation administrator.

--ERROR JM 4550-- User not valid for access to interactive services.

Condition Identifier: JME\$INTERACTIVE_ACCESS_DENIED Code: JM 4550

Description: Your user name is not validated for interactive access to the system. Therefore, you cannot log into the system using an interactive terminal.

User Action: If you have a requirement for interactive services, contact your validation administrator.

--ERROR JM 4555-- There are jobs in the queues that cannot be resubmitted because one or more job levelers are not responding.

Condition Identifier: JME\$LEVELER_NOT_RESPONDING Code: JM 4555

Description: A job leveler task has not cleaned up as required for activation of a profile.

User Action: Attempt the profile activation again after a leveler cycle completes. If the condition persists after several leveler cycles, contact a site analyst. The file server may be failing or a machine may be hung.

--ERROR JM 6005-- End of HISTORY log reached.

Condition Identifier: JME\$END_OF_LOG_REACHED Code: JM 6005

Description: The beginning statistic of the job, JOB_QUEUEING_STARTED, was not located in the log after the position specified on the BEGINNING_LOG_POSITION parameter.

User Action: Check the BEGINNING_LOG_POSITION value on the same command.

--ERROR JM 6010-- No space for ALLOCATE.

Condition Identifier: JME\$NO_SPACE_FOR_ALLOCATE Code: JM 6010

Description: An internal error has occurred while processing a job's history.

User Action: Notify a site analyst.

--ERROR JM 6015-- Internal error processing HISTORY.

Condition Identifier: JME\$JH_INTERNAL_ERROR Code: JM 6015

Description: An internal error occurred while processing a job's history.

User Action: Notify a site analyst and submit \$HISTORY_LOG with a PSR.

--ERROR JM 6020-- Requested job not found in HISTORY log.

Condition Identifier: JME\$JH_NO_JOBS_TO_DISPLAY Code: JM 6020

Description: The job_name on the JOB_NAME parameter was not found in the \$HISTORY_LOG.

User Action: Verify that the correct job_name was specified on the command.

--ERROR JM 6025-- Job History is not activated in the system.

Condition Identifier: JME\$JH_JOB_HISTORY_NOT_ACTIVE Code: JM 6025

Description: Job history logging must be activated by a system operator before job and output history events can be logged.

User Action: Notify a site analyst.

--WARNING JM 7001-- No valid profile was found, standard profile activated.

Condition Identifier: JME\$UNABLE_TO_RECOVER_PROFILE Code: JM 7001

Description: The scheduling profile could not be recovered.

User Action: Recreate the scheduling profile.

--ERROR JM 7002-- Activating profile will cause jobs to be resubmitted.

Condition Identifier: JME\$MAJOR_PROFILE_CHANGE Code: JM 7002

Description: A change in the scheduling profile that may cause the class membership of existing jobs to change has occurred and ENABLE_JOB_RECLASSIFICATION= FALSE.

User Action: Re-enter the command with ENABLE_JOB_RECLASSIFICATION=TRUE, or wait until the jobs have terminated to activate the profile.

--ERROR JM 7003-- The system tables do not match the current profile.

Condition Identifier: JME\$SYSTEM_PROFILE_MISMATCH Code: JM 7003

Description: The profile identification, the job class names, or the service class names in the profile do not match the scheduler tables.

User Action: Re-create the scheduling profile.

--ERROR JM 7004-- The profile contains too many {text} to be activated. Deadstart selections allow a maximum of {text} and there are {text} on the profile.

Condition Identifier: JME\$PROFILE_IS_TOO_LARGE Code: JM 7004

Description: The number of job classes, service classes, and/or applications in the profile is too large.

User Action: Remove some job classes, service classes, and/or applications from the profile; or increase the value of the system attributes MAXIMUM_JOB_CLASSES and MAXIMUM_SERVICE_CLASSES at the next deadstart and then activate the profile.

--ERROR JM 7006-- The profile on {file} does not contain controls for mainframe {text}.

Condition Identifier: JME\$NO_CONTROLS_FOR_MAINFRAME Code: JM 7006

Description: The profile does not contain controls for this mainframe.

User Action: Add controls for this mainframe or use the correct profile.

--WARNING JM 7007-- The scheduler tables were updated but the system scheduler profile file was not.

Condition Identifier: JME\$UPDATED_ONLY_TABLES Code: JM 7007

Description: A new profile file cycle was not written.

User Action: None.

--ERROR JM 7021-- Specifying the key {text} for parameter {text} is not meaningful.

Condition Identifier: JME\$INVALID_ATTRIBUTE_KEYWORD Code: JM 7021

Description: The value specified is not a keyword for the specified parameter.

User Action: Recheck the parameter and keyword that were supplied.

--ERROR JM 7022-- The value specified for the parameter {text} request is invalid.

Condition Identifier: JME\$INVALID_ATTRIBUTE_VALUE Code: JM 7022

Description: The value supplied is not appropriate for the specified parameter.

User Action: Recheck the parameter and value that were supplied.

--ERROR JM 7023-- Expecting {text} value for parameter {text}.

Condition Identifier: JME\$WRONG_KIND_OF_VALUE Code: JM 7023

Description: The value supplied is not appropriate for the specified parameter.

User Action: Recheck the supplied parameter and value.

--ERROR JM 7024-- Specifying the key {text} for {text} of parameter {text} is not meaningful.

Condition Identifier: JME\$INVALID_SUBATTR_KEYWORD Code: JM 7024

Description: The value supplied is not a keyword for the specified parameter.

User Action: Recheck the specified parameter and keyword.

--ERROR JM 7025-- The value specified for {text} of parameter {text} request is invalid.

Condition Identifier: JME\$INVALID_SUBATTRIBUTE_VALUE Code: JM 7025

Description: The value supplied is not appropriate for the specified parameter.

User Action: Recheck the specified parameter and value.

--ERROR JM 7026-- Expecting {text} value for {text} of parameter {text}.

Condition Identifier: JME\$WRONG_KIND_OF_SUBVALUE Code: JM 7026

Description: The value supplied is not appropriate for the specified parameter.

User Action: Recheck the specified parameter and value.

--ERROR JM 7027-- The value specified for {text} of parameter {text} is out of range.

Condition Identifier: JME\$SUBVALUE_OUT_OF_RANGE Code: JM 7027

Description: The value specified is not in the acceptable range.

User Action: Recheck the parameter and value that were supplied.

--ERROR JM 7041-- {text} is not a valid attribute.

Condition Identifier: JME\$UNKNOWN_ATTRIBUTE Code: JM 7041

Description: The attribute specified is not known to the utility.

User Action: Recheck the attribute that was supplied.

--ERROR JM 7042-- {text} is not a valid group name.

Condition Identifier: JME\$UNKNOWN_GROUP_NAME Code: JM 7042

Description: The group name specified is not known to the utility.

User Action: Recheck the group name that was supplied.

--ERROR JM 7045-- {text} is undefined.

Condition Identifier: JME\$CURRENT_OBJECT_UNDEFINED Code: JM 7045

Description: The object of the function is undefined.

User Action: Submit a PSR.

--ERROR JM 7061-- Cannot read profile {file}.

Condition Identifier: JME\$CANNOT_READ_PROFILE Code: JM 7061

Description: The contents of the profile have been damaged.

User Action: Recreate the profile.

--ERROR JM 7081-- The {text} {text} is not known.

Condition Identifier: JME\$OBJECT_NOT_KNOWN Code: JM 7081

Description: The object specified on the command is not known to the utility.

User Action: Recheck the specified object.

--ERROR JM 7082-- The {text} {text} already exists.

Condition Identifier: JME\$OBJECT_ALREADY_KNOWN Code: JM 7082

Description: The object to be created already exists.

User Action: Recheck the specified object.

--ERROR JM 7083-- Adding {text} exceeds the limit for {text}.

Condition Identifier: JME\$TOO_MANY_OBJECTS Code: JM 7083

Description: Adding this object exceeds the maximum number of objects allowed on a profile.

User Action: Rework your scheduling plan to require fewer objects.

--ERROR JM 7084-- The {text} UNASSIGNED cannot be moved.

Condition Identifier: JME\$CANNOT_MOVE_UNASSIGNED Code: JM 7084

Description: The SELECTION_RANK attribute of class UNASSIGNED cannot be changed.

User Action: Do not try to change the SELECTION_RANK attribute of class UNASSIGNED.

--ERROR JM 7085-- Duplicate categories in the excluded and required lists for {text} {text}.

Condition Identifier: JME\$DUPLICATE_CATEGORIES Code: JM 7085

Description: A category cannot be both excluded and required.

User Action: Recheck the EXCLUDED_CATEGORIES and REQUIRED_CATEGORIES.

--ERROR JM 7086-- The abbreviation {text} is already being used.

Condition Identifier: JME\$DUPLICATE_ABBREVIATION Code: JM 7086

Description: The abbreviation given for the class is being used for another class.

User Action: Choose a different abbreviation.

--ERROR JM 7087-- The abbreviation may be changed for only one object at a time.

Condition Identifier: JME\$ABBREVIATION_CHANGE_ILLEGAL Code: JM 7087

Description: Only one class name may be specified if the abbreviation attribute is being changed.

User Action: Enter a separate subcommand for each class whose abbreviation you wish to change.

--ERROR JM 7088-- Cannot delete {text} since a class or category references it.

Condition Identifier: JME\$PROFILE_OBJECT_REFERENCED Code: JM 7088

Description: The object you are trying to delete is referenced elsewhere in the profile.

User Action: Delete all references to the object before deleting the object itself.

--ERROR JM 7089-- Cannot delete {text} {text}.

Condition Identifier: JME\$PERMANENT_OBJECT Code: JM 7089

Description: The object specified cannot be deleted.

User Action: Recheck the object to be deleted.

--ERROR JM 7090-- The {text} {text} is missing from the list.

Condition Identifier: JME\$ITEM_TO_DELETE_IS_MISSING Code: JM 7090

Description: The item being deleted is not present in the list.

User Action: Recheck the item to be deleted.

--ERROR JM 7091-- The {text} {text} already is present in the list.

Condition Identifier: JME\$ITEM_TO_ADD_IS_PRESENT Code: JM 7091

Description: The item being added is already in the list.

User Action: Recheck the item to be added.

--ERROR JM 7092-- The minimum percentages cannot total more than 100 percent for the CPU_Dispatching_Allocation attribute.

Condition Identifier: JME\$PERCENT_SUMS_OVER_100 Code: JM 7092

Description: The sums of the minimum percentage field for each dispatching priority within the CPU_DISPATCHING_ALLOCATION attribute exceed 100 percent.

User Action: Choose new minimum percentages such that the total is less than or equal to 100 percent.

--ERROR JM 7101-- The attributes of this job prevent it from being assigned to the Job Class {text}.

Condition Identifier: JME\$CANNOT_ASSIGN_TO_JOB_CLASS Code: JM 7101

Description: One or more attributes of your job is incompatible with your site's definition of the job class.

User Action: Either revise the SUBMIT_JOB, JOB, or LOGIN statement to allow the job to run in the job class, choose a different job class, or inform a site analyst that a problem exists.

--ERROR JM 7102-- The required combination of resources needed to run this job is unavailable.

Condition Identifier: JME\$NO_MAINFRAME_FOUND_FOR_JOB Code: JM 7102

Description: The job requires resources that are either not available or not available on a single mainframe.

User Action: Inform a site analyst that the required combination of resources was not found for the job.

--ERROR JM 7103-- The attributes of this job prevent it from being a member of any Job Class.

Condition Identifier: JME\$NO_JOB_CLASS_FOUND_FOR_JOB Code: JM 7103

Description: The job was submitted with JOB_CLASS=AUTOMATIC, and no job classes were found which both allow automatic class selection and are compatible with the job attributes.

User Action: Either revise the SUBMIT_JOB, JOB, or LOGIN statement so the job fits one of the automatic job selection classes, or inform a site analyst of the problem.

--ERROR JM 7104-- The Job Class {text} does not exist.

Condition Identifier: JME\$JOB_CLASS_DOES_NOT_EXIST Code: JM 7104

Description: You are validated for the specified job class but it does not exist.

User Action: Consult a site analyst.

--ERROR JM 7120-- Internal profile error {text}.

Condition Identifier: JME\$PROFILE_INTERNAL_ERROR Code: JM 7120

Description: An internal error occurred in the scheduling utility.

User Action: Submit a PSR.

--ERROR KT 1-- {text} is not a recognized KERMIT-VE command. Enter a valid command or type HELP.

Condition Identifier: KTE\$INVALID_COMMAND Code: KT 1

Description: The command that was entered may have been mistyped or is not in the KERMIT command list.

User Action: Check for spelling errors or type HELP to get a list of valid KERMIT commands.

--ERROR KT 2-- A parameter is required for: {text}

Condition Identifier: KTE\$REQUIRED_PARM_MISSING Code: KT 2

Description: The command entered requires additional information in the form of a parameter in order to be executed.

User Action: Enter the command again with a valid parameter value. Command parameters can be found in the online manual by requesting the appropriate command index.

--ERROR KT 3-- You have improper access to the online manual specified. Ensure that the pathname is correct and that you have read and execute access. Pathname specified: {text}

Condition Identifier: KTE\$NO_ONLINE_MANUAL Code: KT 3

Description: The manual cannot be opened and executed for some reason, possibly due to a damaged segment, access problem, or not being available.

User Action: This error will not normally appear during KERMIT use. If it does, then the manual pathname is incorrect in the build or you do not have proper access. Contact system support personnel.

--ERROR KT 4-- Value is outside of the valid range. Valid values are: {text}

Condition Identifier: KTE\$OUT_OF_BOUNDS Code: KT 4

Description: The value specified for a given parameter is not within the valid range.

User Action: Check the range given in the error message and re-enter the command with a valid parameter value.

--ERROR KT 5-- The eight bit prefix character must be different from the control quoting character. {text}

Condition Identifier: KTE\$QUOTE_AND_EIGHT_EQUAL Code: KT 5

Description: You have attempted to set the control quoting character and the eight bit prefix character to the same value. This is not allowed.

User Action: No action is required. If the 8-bit or quote character must be changed, specify a different printable character for that parameter.

--WARNING KT 6-- The filename specified is already open: {text}

Condition Identifier: KTE\$FILE_ALREADY_OPEN Code: KT 6

Description: You have attempted to open a log file that is already open. If this is done, the information in the current log file will be lost.

User Action: Specify a different log filename. The first log file will be closed and the log messages will be routed to the new file.

--ERROR KT 7-- There is no error message associated with this condition. Contact your system support personnel.

Condition Identifier: KTE\$UNDEFINED_ERROR Code: KT 7

Description: This message is triggered when KERMIT enters an unexpected and, therefore, undefined state. This is a program error.

User Action: This error should not appear during KERMIT use. It is designed to trap all unexpected errors encountered during program development. Contact your system support personnel.

--ERROR LI 0-- + Lisp system abort due to internal error: {text} .

Condition Identifier: LIE\$LISP_SYSTEM_ABORT Code: LI 0

Description: LISP internal error.

User Action: Report to your site analyst.

--ERROR LL 0-- {text} load prematurely terminated. Consult load map and/or output.

Condition Identifier: LLE\$PREMATURE_LOAD_TERMINATION Code: LL 0

Description: The loader could not complete the load.

User Action: Determine the cause of the loader error from the diagnostics on the load map. If the load request specified no load map, reattempt the load in order to get a load map.

--ERROR LL 1-- Load error severity is or exceeds TERMINATION_ERROR_LEVEL.

Condition Identifier: LLE\$TERM_ERROR_LEVEL_EXCEEDED Code: LL 1

Description: The load attempt failed.

User Action: Determine the cause of the load error by referring to the diagnostics written on the load map.

--WARNING LL 2-- Load map suspended due to file access error.

Condition Identifier: LLE\$LOAD_MAP_SUSPENDED Code: LL 2

Description: The loader could not write the load map because it could not open the load map file.

User Action: Either ensure that the loader can access the load map file or specify another file as the load map file.

--ERROR LL 3-- Unable to find entry point ({text}).

Condition Identifier: LLE\$ENTRY_POINT_NOT_FOUND Code: LL 3

Description: The files specified on the load request do not contain a required entry point.

User Action: Ensure that all required files are specified on the load request.

--ERROR LL 4-- Maximum segment length exceeded. Consult load map and/or output.

Condition Identifier: LLE\$INSUFFICIENT_MEMORY_TO_LOAD Code: LL 4

Description: One or more sections or segments in the module to be loaded is larger than the system attribute MAXIMUM_SEGMENT_LENGTH, which can be changed by your site.

User Action: Regenerate the object module by recompiling the source code with the problem sections or segments reduced in size. If necessary, consult the site analyst.

--ERROR LL 5-- Loader malfunctioned - {text}. Consult load map and/or output.

Condition Identifier: LLE\$LOADER_MALFUNCTIONED Code: LL 5

Description: An internal loader error has occurred.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR LL 100-- Unsatisfied reference to external {text}. Referenced by:

Condition Identifier: LLE\$UNSATISFIED_EXTERNAL Code: LL 100

Description: The specified files do not contain a referenced entry point.

User Action: Ensure that all required files are specified on the load request.

--FATAL LL 101-- Starting procedure not specified.

Condition Identifier: LLE\$TRANSFER_SYMBOL_MISSING Code: LL 101

Description: The modules loaded do not contain a transfer symbol record and no starting procedure was specified on the load request.

User Action: Either specify the starting procedure or ensure that the modules loaded include a transfer symbol record. A compiler generates a transfer symbol record when it processes a PROGRAM statement indicating the starting point of a program.

--FATAL LL 102-- Starting procedure {text} not found.

Condition Identifier: LLE\$TRANSFER_SYMBOL_UNDEFINED Code: LL 102

Description: The files specified on the load request do not contain the entry point specified as the starting procedure.

User Action: Either change the starting procedure name or ensure that the module containing the starting procedure is loaded.

--FATAL LL 103-- Program transfer address not aligned on word boundary.

Condition Identifier: LLE\$TRANSFER_SYMBOL_UNALIGNED Code: LL 103

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--INFORMATIVE LL 104-- No entry point cross references detected.

Condition Identifier: LLE\$NO_XREFS_TO_REPORT Code: LL 104

Description: The load required no external reference satisfaction so no external references are listed on the load map.

User Action: None.

--ERROR LL 105-- Module {text} not found.

Condition Identifier: LLE\$MODULE_NOT_FOUND Code: LL 105

Description: The loader could not find a module specified in the module list within the files in the program library list.

User Action: Ensure that the module is specified correctly and ensure that the file containing the module is specified on the load request.

--ERROR LL 110-- File {file} is not a library.

Condition Identifier: LLE\$FILE_NOT_LIBRARY Code: LL 110

Description: The specified file must be an object library.

User Action: Ensure that the file is an object library and that it is specified correctly.

--FATAL LL 111-- Library header missing from library {file}.

Condition Identifier: LLE\$LIBRARY_HEADER_MISSING Code: LL 111

Description: The header is missing from the object library file.

User Action: Regenerate the object library.

--FATAL LL 112-- Empty module dictionary in library {file}.

Condition Identifier: LLE\$EMPTY_MODULE_DICTIONARY Code: LL 112

Description: The object library contains no modules.

User Action: Regenerate or omit the object library.

--FATAL LL 113-- Erroneous pointer to module dictionary in library {file}.

Condition Identifier: LLE\$BAD_MODULE_DICTIONARY_PTR Code: LL 113

Description: The object library structure is in error.

User Action: Regenerate the object library. For information on creating an object library, refer to the NOS/VE Object Code Management manual.

--FATAL LL 115-- Erroneous pointer to entry point dictionary in library {file}.

Condition Identifier: LLE\$BAD_ENTRY_DICTIONARY_PTR Code: LL 115

Description: The object library structure is in error.

User Action: Regenerate the object library.

--FATAL LL 116-- Erroneous pointer to module header from {text} dictionary entry {text} in library {file}.

Condition Identifier: LLE\$BAD_MODULE_HEADER_PTR Code: LL 116

Description: The object library structure is in error.

User Action: Regenerate the object library.

--FATAL LL 117-- Erroneous pointer to allotted section from module {text} at offset {text}.

Condition Identifier: LLE\$BAD_ALLOTTED_SECTION_PTR Code: LL 117

Description: The module structure is in error.

User Action: Regenerate the module by recompiling the source code, and regenerate the object library.

--FATAL LL 118-- Erroneous pointer to interpretive element from module header at offset {text} in library {file}.

Condition Identifier: LLE\$BAD_INTERPRETIVE_ELEM_PTR Code: LL 118

Description: The object library structure is in error.

User Action: Regenerate the module. To regenerate the object module, recompile the source code. To regenerate a load module, regenerate the object library. For information on creating an object library, refer to the NOS/VE Object Code Management manual.

--FATAL LL 119-- Object library version must be '{text}' for library {file}.

Condition Identifier: LLE\$WRONG_LIBRARY_VERSION Code: LL 119

Description: The loader cannot use the specified version of the object library.

User Action: Regenerate the object library using the current version of the object library generator.

--FATAL LL 120-- Identification record expected at offset {text} in file {file}.

Condition Identifier: LLE\$IDENTIFICATION_EXPECTED Code: LL 120

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 121-- Unknown record kind encountered at offset {text}.

Condition Identifier: LLE\$UNKNOWN_RECORD_KIND Code: LL 121

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 122-- Invalid fixer value in record at offset {text}.

Condition Identifier: LLE\$BAD_FIXER_VALUE Code: LL 122

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 123-- PPU absolute record encountered at offset {text}.

Condition Identifier: LLE\$PPU_ABSOLUTE_ENCOUNTERED Code: LL 123

Description: The module format appears to contain PPU code.

User Action: Ensure that the module specified contains CPU code.

--FATAL LL 124-- Premature EOI encountered on file {file}. Last valid record at offset {text}.

Condition Identifier: LLE\$PREMATURE_EOF Code: LL 124

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 126-- No transfer symbol record encountered for module {text}.

Condition Identifier: LLE\$TRANSFER_RECORD_MISSING Code: LL 126

Description: No starting procedure was found in the specified module.

User Action: Specify a starting procedure on the EXECUTE_TASK command.

--WARNING LL 130-- Unknown date format in identification record at offset {text}.

Condition Identifier: LLE\$UNKNOWN_DATE_FORMAT Code: LL 130

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 131-- Object text version must be '{text}', but is '{text}'.

Condition Identifier: LLE\$WRONG_OBJECT_TEXT_VERSION Code: LL 131

Description: The loader cannot use the object module version.

User Action: Regenerate the module using the current version of the compiler.

--WARNING LL 132-- Unknown generator specified in identification record at offset {text}.

Condition Identifier: LLE\$UNKNOWN_GENERATOR Code: LL 132

Description: The loader did not recognize the code generator for the module.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 133-- Module kind is not virtual state in identification record at offset {text}.

Condition Identifier: LLE\$MODULE_WRONG_KIND Code: LL 133

Description: An attempt was made to load a module containing vector virtual state instructions on a mainframe module not capable of executing this type of instruction.

User Action: Recompile module without the vector virtual state capability included.

--FATAL LL 134-- Nonexecutable module specified by identification record at offset {text}.

Condition Identifier: LLE\$MODULE_NONEXECUTABLE Code: LL 134

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code. Make sure no compile errors are generated.

--ERROR LL 140-- Duplicate section definition encountered at offset {text}.

Condition Identifier: LLE\$DUPLICATE_SECTION_DEF Code: LL 140

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 141-- Unknown section kind in section definition record at offset {text}.

Condition Identifier: LLE\$UNKNOWN_SECTION_KIND Code: LL 141

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--WARNING LL 142-- Binding section definition record at offset {text} contains access attribute other than BINDING.

Condition Identifier: LLE\$IMPROPER_B_S_ATTRIBUTES Code: LL 142

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--WARNING LL 143-- BINDING access attribute ignored in section definition record at offset {text}.

Condition Identifier: LLE\$BINDING_ATTR_NOT_ALLOWED Code: LL 143

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 144-- Both WRITE and EXECUTE access attributes specified in section definition record at offset {text}.

Condition Identifier: LLE\$WRITE_EXECUTE_SECTION Code: LL 144

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 145-- Allocation alignment of zero specified in section definition record at offset {text}.

Condition Identifier: LLE\$SECTION_ALIGNMENT_ZERO Code: LL 145

Description: An object module was generated containing an invalid object code.

User Action: If the problem recurs, inform the site analyst.

--FATAL LL 146-- Code element missing for section definition record at offset {text}.

Condition Identifier: LLE\$CODE_ELEMENT_MISSING Code: LL 146

Description: The module format is in error.

User Action: Regenerate the module by recompiling the source code.

--FATAL LL 147-- Binding section definition record at offset {text} must cause section to be aligned on a word boundary.

Condition Identifier: LLE\$BINDING_SECTION_UNALIGNED Code: LL 147

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 149-- Attempt by {text} at offset {text} to initialize locations in unallocated common.

Condition Identifier: LLE\$INVALID_TEXT_IN_COMMON Code: LL 149

Description: The module contains a record type at the offset shown that attempts to initialize locations in an unallocated common block.

User Action: Recompile the module without the statements that initialize the common block.

--ERROR LL 150-- Section ordinal in {text} at offset {text} exceeds greatest section ordinal defined for module.

Condition Identifier: LLE\$INVALID_SECTION_ORDINAL Code: LL 150

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 151-- Undefined section referenced by {text} at offset {text}.

Condition Identifier: LLE\$UNDEFINED_SECTION Code: LL 151

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 152-- Attempt by {text} at offset {text} to initialize locations beyond section boundary.

Condition Identifier: LLE\$INVALID_SECTION_OFFSET Code: LL 152

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 153-- Unknown linkage kind specified in {text} at offset {text}.

Condition Identifier: LLE\$UNKNOWN_ADDRESS_KIND Code: LL 153

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--WARNING LL 154-- Unknown language specified in {text} at offset {text}.

Condition Identifier: LLE\$UNKNOWN_LANGUAGE Code: LL 154

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 155-- Linkage specified by {text} at offset {text} not aligned on word boundary.

Condition Identifier: LLE\$IMPROPER_LINKAGE_ALIGNMENT Code: LL 155

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 156-- Attempt to generate non-linkage data in binding section by {text} at offset {text}.

Condition Identifier: LLE\$NON_LINKAGE_BINDING_DATA Code: LL 156

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 157-- Destination of external linkage item, {text} - offset {text}, of module {text} is a code or an allotted section.

Condition Identifier: LLE\$IMPROPER_LINKAGE_ITEM Code: LL 157

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 160-- Binding section overwrite attempted by linkage to entry point {text} from module {text}.

Condition Identifier: LLE\$BINDING_SECTION_OVERWRITE Code: LL 160

Description: The module format is in error.

User Action: Regenerate the module by recompiling the source code.

--ERROR LL 161-- Declaration mismatch on reference to entry point {text} from module {text}, using source text checking.

Condition Identifier: LLE\$DECLARATION_MISMATCH Code: LL 161

Description: The formal and actual parameters on a call do not match.

User Action: Compare the source text for the two modules to ensure that the entry point and external reference match. Depending on the difference between the entry point and external reference declarations, you may have to recompile your program, or you may be able to suppress the error by setting `TERMINATION_ERROR_LEVEL=FATAL`.

--ERROR LL 162-- Entry point {text} must be aligned on word boundary for procedure reference from module {text}.

Condition Identifier: LLE\$ENTRY_POINT_UNALIGNED Code: LL 162

Description: The module format is in error.

User Action: Regenerate the module by recompiling the source code.

--FATAL LL 164-- Declaration mismatch on reference to entry point {text} from module {text}, using object text checking.

Condition Identifier: LLE\$F_DECLARATION_MISMATCH Code: LL 164

Description: The loader detected a structural mismatch between an entry point declaration and its external reference.

User Action: Be sure you specified the correct types of parameters in the correct order. Recompile your program using the same type declarations that were used when compiling the module containing the entry point which caused the declaration mismatch.

--ERROR LL 170-- Duplicate definition of entry point {text} encountered. First definition holds.

Condition Identifier: LLE\$DUPLICATE_ENTRY_POINT Code: LL 170

Description: The specified entry point has already been loaded.

User Action: Eliminate the duplication.

--ERROR LL 180-- Bit string size > 63 or bit string offset not in range (1..7) in bit string insertion record at offset {text}.

Condition Identifier: LLE\$INVALID_BIT_STRING Code: LL 180

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 181-- Bit string span of more than 8 bytes specified in bit string insertion record at offset {text}.

Condition Identifier: LLE\$INVALID_BIT_STRING_SPAN Code: LL 181

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--FATAL LL 190-- Binding section overwrite attempted by address formulation record at offset {text}.

Condition Identifier: LLE\$ADD_FORM_B_S_OVERWRITE Code: LL 190

Description: The module format is in error.

User Action: Regenerate the module by recompiling the source code.

--ERROR LL 191-- Target of procedure reference generated by address formulation item at offset {text} not aligned on word boundary.

Condition Identifier: LLE\$VALUE_ADDRESS_UNALIGNED Code: LL 191

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--ERROR LL 192-- Destination of address formulation item, offset {text}, is a code or an allotted section.

Condition Identifier: LLE\$IMPROPER_ADD_FORM_ITEM Code: LL 192

Description: The module format is in error.

User Action: Regenerate the object module by recompiling the source code.

--WARNING LL 200-- Size of extensible section exceeds program segment size. Section truncated.

Condition Identifier: LLE\$EXTENSIBLE_TRUNCATED Code: LL 200

Description: The program requested blank common greater than the maximum program segment size for which the user is validated. The blank common is truncated to the maximum program segment size.

User Action: Reduce the amount of blank common requested by the program.

--WARNING LL 201-- Size of extensible common block {text} greater than program segment size. Common block truncated.

Condition Identifier: LLE\$EXTENSIBLE_COMMON_TRUNCATED Code: LL 201

Description: The program requested a common block greater than the maximum program segment size for which the user is validated. The common block is truncated to the maximum program segment size.

User Action: Reduce the size of the specified common block.

--ERROR LL 202-- Size of common block {text} greater than previous definition. Common block truncated.

Condition Identifier: LLE\$COMMON_TRUNCATED Code: LL 202

Description: The first common block encountered determines the size of the common block.

User Action: Increase the size of the first common block.

--WARNING LL 203-- Size of common block {text} less than previous definition.

Condition Identifier: LLE\$COMMON_SIZE_MISMATCH Code: LL 203

Description: The first COMMON encountered defines the length of the COMMON block.

User Action: Change one of the common block lengths and regenerate the object module by recompiling the source code.

--ERROR LL 204-- Access attributes for common block {text} do not match previous definition.

Condition Identifier: LLE\$COMMON_ATTR_MISMATCH Code: LL 204

Description: The common block definitions for two modules do not match.

User Action: Collect the definition in one of the modules and regenerate the object module by recompiling the source code.

--ERROR LL 210-- File {file} is empty.

Condition Identifier: LLE\$EMPTY_LOAD_FILE Code: LL 210

Description: The specified file does not contain object modules or an object library.

User Action: Ensure that the correct file is specified for the program.

--FATAL LL 211-- File {file} does not have proper attributes (file_contents/file_structure) must be: (OBJECT/DATA) or (OBJECT/LIBRARY)

Condition Identifier: LLE\$FILE_NOT_LOAD_FILE Code: LL 211

Description: The loader does not recognize the file as an object file or object library file because its file_contents and file_structure attribute values are incorrect. A load file is a file generated by a compiler or the object library generator; the compiler or object library generator sets these attribute values.

User Action: Ensure that a load file has been specified and that its file attribute values have not been changed.

--FATAL LL 212-- Unable to access load file -- file ignored.

Condition Identifier: LLE\$UNABLE_TO_ACCESS_LOAD_FILE Code: LL 212

Description: The loader could not access the specified file.

User Action: Ensure that the file is specified correctly and that the loader can read it.

--FATAL LL 213-- Unable to load file, {file}, containing FAP -- file ignored.

Condition Identifier: LLE\$FILE_CONTAINS_FAP Code: LL 213

Description: The loader could not load the file containing the file access procedure for a file.

User Action: Ensure that the file is accessible and that it is named within the program description.

--ERROR LL 216-- The number of parameters do not match in the call to procedure {text} from module {text}.

Condition Identifier: LLE\$INVALID_MATCHING Code: LL 216

Description: The number of parameters is incompatible between a formal procedure statement and a call to it.

User Action: Change program source code so the number of parameters in the procedure call matches the number of parameters in the procedure statement. Recompile.

--FATAL LL 220-- Premature EOF encountered on module {text}, last valid record found at offset {text}.

Condition Identifier: LLE\$PREMATURE_EOF_ON_MODULE Code: LL 220

Description: The module format is in error.

User Action: Regenerate the object module by recompiling its source code. If the error recurs, notify a site analyst.

--ERROR LL 378-- {text} {text} has an integer length mismatch.

Condition Identifier: LLE\$BAD_INTEGER_LENGTH Code: LL 378

Description: Lengths of parameters are incompatible.

User Action: Change program source code so parameters have compatible lengths. Recompile.

--ERROR LL 379-- {text} {text} has an array size mismatch.

Condition Identifier: LLE\$INVALID_ARRAY_SIZE_MATCHING Code: LL 379

Description: Parameter array sizes are incompatible.

User Action: Change program source code so parameters have compatible array sizes. Recompile.

--CATASTROPHIC LL 380-- Overflow of program segment {text}.

Condition Identifier: LLE\$PROGRAM_SEGMENT_OVERFLOW Code: LL 380

Description: The program is larger than the maximum program segment size allowed the user.

User Action: Reduce the size of the program.

--CATASTROPHIC LL 381-- Unable to create program segment {text}.

Condition Identifier: LLE\$UNABLE_TO_CREATE_PROG_SEG Code: LL 381

Description: The program is already using the maximum number of segments allowed the user.

User Action: Reduce the size of the program so that another segment is not needed.

--CATASTROPHIC LL 382-- Unable to determine size of program segment {text}.

Condition Identifier: LLE\$UNABLE_TO_GET_PROG_SEG_SIZE Code: LL 382

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 383-- Unable to fix attributes of program segment {text}.

Condition Identifier: LLE\$UNABLE_TO_FIX_PROG_SEG_ATTR Code: LL 383

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 384-- Unable to defix attributes of program segment {text}.

Condition Identifier: LLE\$UNABLE_TO_DEFIX_SEG_ATTR Code: LL 384

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 385-- Unable to create {text} container.

Condition Identifier: LLE\$UNABLE_TO_CREATE_TABLE Code: LL 385

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 386-- Overflow of {text} container.

Condition Identifier: LLE\$LOADER_TABLE_OVERFLOW Code: LL 386

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 388-- End of file encountered on APD file {file}.

Condition Identifier: LLE\$EOF_ENCOUNTERED_ON_APD_FILE Code: LL 388

Description: The internal statistics file processed by the program measurement utility is in error.

User Action: If you are using a saved program environment, ensure that the file data has not been changed.
Otherwise, notify a site analyst, providing appropriate documentation.**--CATASTROPHIC LL 389-- Local block name {text} conflicts with existing APD block name map.**

Condition Identifier: LLE\$BAD_LOCAL_BLOCK_NAME Code: LL 389

Description: Internal program measurement utility error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 390-- Remote block name {text} conflicts with APD block name map.

Condition Identifier: LLE\$BAD_REMOTE_BLOCK_NAME Code: LL 390

Description: Internal program measurement utility error.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 391-- Unable to access Analyze Program Dynamics file {file}.

Condition Identifier: LLE\$UNABLE_TO_ACCESS_APD_FILE Code: LL 391

Description: The internal statistics file processed by the program measurement utility is in error.

User Action: If you are using a saved program environment, ensure that the file data has not been changed. Otherwise, notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 392-- System error - Loader stopped - See Site Analyst.

Condition Identifier: LLE\$LOADER_STOPPED Code: LL 392

Description: An internal loader error has occurred.

User Action: Notify a site analyst, providing appropriate documentation.

--CATASTROPHIC LL 394-- Unable to move text to byte offset {text}.

Condition Identifier: LLE\$UNABLE_TO_MOVE_TEXT Code: LL 394

Description: Internal loader error.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR LL 395-- {text} {text} has a type mismatch.

Condition Identifier: LLE\$INVALID_TYPE_MATCHING Code: LL 395

Description: Call parameter types do not match.

User Action: Recompile module(s) specifying matching parameter types.

--ERROR LL 396-- {text} {text} has a kind mismatch.

Condition Identifier: LLE\$INVALID_KIND_MATCHING Code: LL 396

Description: Call parameter kinds do not match.

User Action: Recompile module(s) specifying matching parameter kinds.

--ERROR LL 397-- {text} {text} has a mode mismatch.

Condition Identifier: LLE\$INVALID_MODE_MATCHING Code: LL 397

Description: Call parameter modes do not match.

User Action: Recompile module(s) specifying matching parameter modes.

--ERROR LL 398-- {text} parameter number {text}

Condition Identifier: LLE\$ACTUAL_LESS_THAN_FORMAL Code: LL 398

Description: The actual parameter named in a call is smaller in length than the formal parameter specified in the callee.

User Action: Recompile, specifying a suitable parameter length.

--ERROR LL 399-- {text} has character length mismatch on parameter {text}.

Condition Identifier: LLE\$BAD_CHAR_LENGTH Code: LL 399

Description: The character lengths of the actual parameter and the formal parameter of a call are not compatible.

User Action: Recompile, specifying a suitable parameter.

--ERROR LL 400-- Erroneous pointer to program header in library.

Condition Identifier: LLE\$BAD_PROGRAM_HEADER_PTR Code: LL 400

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 401-- Erroneous pointer to SCL header in library.

Condition Identifier: LLE\$BAD_SCL_HEADER_PTR Code: LL 401

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 402-- Erroneous {text} library member header in library.

Condition Identifier: LLE\$BAD_LIBRARY_MEMBER_HEADER Code: LL 402

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 403-- Erroneous {text} object text description in library.

Condition Identifier: LLE\$BAD_OBJECT_TEXT_DESCRIPTION Code: LL 403

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 404-- Erroneous pointer to PPU header in library.

Condition Identifier: LLE\$BAD_PPU_HEADER_PTR Code: LL 404

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 405-- Erroneous {text} load module header in library.

Condition Identifier: LLE\$BAD_LOAD_MODULE_HEADER Code: LL 405

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 406-- Erroneous pointer to load header in library.

Condition Identifier: LLE\$BAD_LOAD_HEADER_PTR Code: LL 406

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 407-- Erroneous {text} bad module in library.

Condition Identifier: LLE\$BAD_MODULE Code: LL 407

Description: The object library structure is in error.

User Action: Regenerate the object library.

--ERROR LL 408-- Erroneous {text} file not a load file.

Condition Identifier: LLE\$FILE_NOT_A_LOAD_FILE Code: LL 408

Description: The named file is not an object file or an object library.

User Action: Use an object file or object library for loading modules.

--FATAL ML 1-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ACOS_ARG_INDEF Code: ML 1

Description: Note: This message is also associated with FORTRAN condition code FL341.

The argument passed to ACOS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 2-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$ACOS_ARG_INF Code: ML 2

Description: Note: This message is also associated with FORTRAN condition code FL342.

The argument passed to ACOS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 3-- {text}({text}). Argument must be in range [-1.0,1.0].

Condition Identifier: MLE\$ACOS_ARG_RANGE Code: ML 3

Description: Note: This message is also associated with FORTRAN condition code FL343.

The argument passed to ACOS was not within the valid range.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect argument was generated.

Further Information: Refer to the description of the ACOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 4-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$AINT_ARG_INDEF Code: ML 4

Description: Note: This message is also associated with FORTRAN condition code FL344.

The argument passed to AINT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 5-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$AINT_ARG_INF Code: ML 5

Description: Note: This message is also associated with FORTRAN condition code FL345.

The argument passed to AINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 6-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ALOG_ARG_INDEF Code: ML 6

Description: Note: This message is also associated with FORTRAN condition code FL346.

The argument passed to ALOG has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 7-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$ALOG_ARG_INF Code: ML 7

Description: Note: This message is also associated with FORTRAN condition code FL347.

The argument passed to ALOG has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 8-- {text}(0.0). Argument must be > 0.0.

Condition Identifier: MLE\$ALOG_ARG_0 Code: ML 8

Description: Note: This message is also associated with FORTRAN condition code FL348.

The argument passed to ALOG must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ALOG routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 9-- {text}({text}). Argument must be > 0.0.

Condition Identifier: MLE\$ALOG_ARG_NEG Code: ML 9

Description: Note: This message is also associated with FORTRAN condition code FL349.

The argument passed to ALOG must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ALOG routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 10-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ALOG10_ARG_INDEF Code: ML 10

Description: Note: This message is also associated with FORTRAN condition code FL350.

The argument passed to ALOG10 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 11-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$ALOG10_ARG_INF Code: ML 11

Description: Note: This message is also associated with FORTRAN condition code FL351.

The argument passed to ALOG10 has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 12-- {text}(0.0). Argument must be > 0.0.

Condition Identifier: MLE\$ALOG10_ARG_0 Code: ML 12

Description: Note: This message is also associated with FORTRAN condition code FL352.

The argument passed to ALOG10 must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ALOG10 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 13-- {text}({text}). Argument must be > 0.0.

Condition Identifier: MLE\$ALOG10_ARG_NEG Code: ML 13

Description: Note: This message is also associated with FORTRAN condition code FL353.

The argument passed to ALOG10 must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ALOG10 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 14-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$AMOD_ARG1_INDEF Code: ML 14

Description: Note: This message is also associated with FORTRAN condition code FL354.

The first argument passed to AMOD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 15-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$AMOD_ARG2_INDEF Code: ML 15

Description: Note: This message is also associated with FORTRAN condition code FL355.

The second argument passed to AMOD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 16-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$AMOD_ARG1_INF Code: ML 16

Description: Note: This message is also associated with FORTRAN condition code FL356.

The first argument passed to AMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the first argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 17-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$AMOD_ARG2_INF Code: ML 17

Description: Note: This message is also associated with FORTRAN condition code FL357.

The second argument passed to AMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 18-- {text}(arg1={text},arg2=0.0). Arg2 must be nonzero.

Condition Identifier: MLE\$AMOD_ARG2_0 Code: ML 18

Description: Note: This message is also associated with FORTRAN condition code FL358.

The second argument passed to AMOD must not be zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the AMOD routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 19-- {text}(arg1={text},arg2={text}). Arg1/arg2 infinite.

Condition Identifier: MLE\$AMOD_ARGS_RANGE Code: ML 19

Description: Note: This message is also associated with FORTRAN condition code FL359.

The result of dividing the arguments passed to AMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 20-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ANINT_ARG_INDEF Code: ML 20

Description: Note: This message is also associated with FORTRAN condition code FL360.

The argument passed to ANINT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 21-- {text}({text}). Arg infinite.

Condition Identifier: MLE\$ANINT_ARG_INF Code: ML 21

Description: Note: This message is also associated with FORTRAN condition code FL361.

The argument passed to ANINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 22-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ASIN_ARG_INDEF Code: ML 22

Description: Note: This message is also associated with FORTRAN condition code FL362.

The argument passed to ASIN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 23-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$ASIN_ARG_INF Code: ML 23

Description: Note: This message is also associated with FORTRAN condition code FL363.

The argument passed to ASIN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 24-- {text}({text}). Argument must be in range [-1.0,1.0].

Condition Identifier: MLE\$ASIN_ARG_RANGE Code: ML 24

Description: Note: This message is also associated with FORTRAN condition code FL364.

The argument passed to ASIN must be in the range from -1.0 to 1.0.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ASIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 25-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ATAN_ARG_INDEF Code: ML 25

Description: Note: This message is also associated with FORTRAN condition code FL365.

The argument passed to ATAN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 26-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$ATAN2_ARG1_INDEF Code: ML 26

Description: Note: This message is also associated with FORTRAN condition code FL366.

The first argument passed to ATAN2 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 27-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$ATAN2_ARG2_INDEF Code: ML 27

Description: Note: This message is also associated with FORTRAN condition code FL367.

The second argument passed to ATAN2 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 28-- {text}(arg1={text},arg2={text}). Both arguments infinite.

Condition Identifier: MLE\$ATAN2_ARGS_INF Code: ML 28

Description: Note: This message is also associated with FORTRAN condition code FL368.

The arguments passed to ATAN2 both have infinite values. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the arguments to determine how the infinite values were generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 29-- {text}(0.0,0.0). One argument must be nonzero.

Condition Identifier: MLE\$ATAN2_ARGS_0 Code: ML 29

Description: Note: This message is also associated with FORTRAN condition code FL369.

The arguments passed to ATAN2 must not both be zero.

User Action: Be sure that you have specified the correct arguments. Then check all previous uses of the arguments to determine how the incorrect value was generated.

Further Information: Refer to the description of the ATAN2 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 30-- {text}(arg1={text},arg2={text}). Arg2 must be zero if arg1/arg2infinite.

Condition Identifier: MLE\$ATAN2_ARGS_RANGE Code: ML 30

Description: Note: This message is also associated with FORTRAN condition code FL370.

The first argument passed to ATAN2 must be zero if the result of arg2/arg1 is infinite. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine if an infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 31-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ATANH_ARG_INDEF Code: ML 31

Description: Note: This message is also associated with FORTRAN condition code FL371.

The argument passed to ATANH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 32-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$ATANH_ARG_INF Code: ML 32

Description: Note: This message is also associated with FORTRAN condition code FL372.

The argument passed to ATANH has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 33-- {text}({text}). ABS(argument) must be < 1.0.

Condition Identifier: MLE\$ATANH_ARG_RANGE Code: ML 33

Description: Note: This message is also associated with FORTRAN condition code FL373.

The absolute value of the argument passed to ATANH must not be less than 1.0.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ATANH routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 34-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CABS_ARG_INDEF Code: ML 34

Description: Note: This message is also associated with FORTRAN condition code FL374.

The argument passed to CABS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 35-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CABS_ARG_INF Code: ML 35

Description: Note: This message is also associated with FORTRAN condition code FL375.

The argument passed to CABS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 36-- {text}({text},{text}). Result infinite.

Condition Identifier: MLE\$CABS_RESULT_INF Code: ML 36

Description: Note: This message is also associated with FORTRAN condition code FL376.

The CABS routine generated an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. The CABS routine calculates the square root of $(\text{real}^2 + \text{imag}^2)$. The probable cause is that either real or imag or both contain a very large value.

User Action: Check the arguments for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, or bad input data.

Further Information: Refer to the description of the CABS function in the FORTRAN Language Definition Usage Manual.

--FATAL ML 37-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CCOS_ARG_INDEF Code: ML 37

Description: Note: This message is also associated with FORTRAN condition code FL377.

The argument passed to CCOS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 38-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CCOS_ARG_INF Code: ML 38

Description: Note: This message is also associated with FORTRAN condition code FL378.

The argument passed to CCOS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 39-- {text}({text},{text}). ABS(real part) must be < 2.47.**

Condition Identifier: MLE\$CCOS_REAL_RANGE Code: ML 39

Description: Note: This message is also associated with FORTRAN condition code FL379.

The absolute value of the real part of the argument passed to CCOS must be less than 2^{47} . ($2^{47} = 1.4074E+14$.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CCOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 40-- {text}({text},{text}). Imag. part must be < 4095.*LOG(2).

Condition Identifier: MLE\$CCOS_IMAG_TOO_BIG Code: ML 40

Description: Note: This message is also associated with FORTRAN condition code FL380.

The absolute value of the imaginary part of the argument passed to CCOS must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CCOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 41-- {text}({text},{text}). Imag. part must be > -4095.*LOG(2).

Condition Identifier: MLE\$CCOS_IMAG_TOO_SMALL Code: ML 41

Description: Note: This message is also associated with FORTRAN condition code FL381.

The imaginary part of the argument passed to CCOS must be greater than -4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CCOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 42-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CEXP_ARG_INDEF Code: ML 42

Description: Note: This message is also associated with FORTRAN condition code FL382.

The argument passed to CEXP has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 43-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CEXP_ARG_INF Code: ML 43

Description: Note: This message is also associated with FORTRAN condition code FL383.

The argument passed to CEXP has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 44-- {text}({text},{text}). ABS(imag. part) must be < 2.47.**

Condition Identifier: MLE\$CEXP_IMAG_RANGE Code: ML 44

Description: Note: This message is also associated with FORTRAN condition code FL384.

The absolute value of the imaginary part of the argument passed to CEXP must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CEXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 45-- {text}({text},{text}). real part must be < 4095.*LOG(2).

Condition Identifier: MLE\$CEXP_REAL_RANGE Code: ML 45

Description: Note: This message is also associated with FORTRAN condition code FL385.

The absolute value of the real part of the argument passed to CEXP must be less than $4095 \cdot \text{LOG}(2)$. ($4095 \cdot \text{LOG}(2)$ is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CEXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 46-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CLOG_ARG_INDEF Code: ML 46

Description: Note: This message is also associated with FORTRAN condition code FL386.

The argument passed to CLOG has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 47-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CLOG_ARG_INF Code: ML 47

Description: Note: This message is also associated with FORTRAN condition code FL387.

The argument passed to CLOG has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 48-- {text}({text},{text}). ABS(argument) infinite.

Condition Identifier: MLE\$CLOG_ABS_ARG_INF Code: ML 48

Description: Note: This message is also associated with FORTRAN condition code FL388.

The absolute value of the argument passed to CLOG has an infinite value. (The absolute value of a complex number is the square root of $(\text{real}^2 + \text{imag}^2)$. One of both components may be in error.)

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 49-- {text}(0.0). One of real or imag. parts must be nonzero.

Condition Identifier: MLE\$CLOG_ARG_0 Code: ML 49

Description: Note: This message is also associated with FORTRAN condition code FL389.

Either the real or imaginary part of the argument passed to CLOG must be nonzero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CLOG routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 50-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$COS_ARG_INDEF Code: ML 50

Description: Note: This message is also associated with FORTRAN condition code FL390.

The argument passed to COS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 51-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$COS_ARG_INF Code: ML 51

Description: Note: This message is also associated with FORTRAN condition code FL391.

The argument passed to COS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 52-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$COS_ARG_RANGE Code: ML 52

Description: Note: This message is also associated with FORTRAN condition code FL392.

The absolute value of the argument passed to COS must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the COS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 53-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$COSH_ARG_INDEF Code: ML 53

Description: Note: This message is also associated with FORTRAN condition code FL393.

The argument passed to COSH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 54-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$COSH_ARG_INF Code: ML 54

Description: Note: This message is also associated with FORTRAN condition code FL394.

The argument passed to COSH has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 55-- {text}({text}). ABS(argument) must be < 4095.*LOG(2).

Condition Identifier: MLE\$COSH_ARG_RANGE Code: ML 55

Description: Note: This message is also associated with FORTRAN condition code FL395.

The absolute value of the argument passed to COSH must be less than $4095 \cdot \text{LOG}(2)$. ($4095 \cdot \text{LOG}(2)$ is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the COSH routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 56-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CSIN_ARG_INDEF Code: ML 56

Description: code FL396.

The argument passed to CSIN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 57-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CSIN_ARG_INF Code: ML 57

Description: Note: This message is also associated with FORTRAN condition code FL397.

The argument passed to CSIN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 58-- {text}({text},{text}). ABS(real part) must be < 2.47.**

Condition Identifier: MLE\$CSIN_REAL_RANGE Code: ML 58

Description: Note: This message is also associated with FORTRAN condition code FL398.

The absolute value of the real part of the argument passed to CSIN must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CSIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 59-- {text}({text},{text}). ABS(imag. part) must be < 4095.*LOG(2).

Condition Identifier: MLE\$CSIN_IMAG_RANGE Code: ML 59

Description: Note: This message is also associated with FORTRAN condition code FL399.

The imaginary part of the argument passed to CSIN must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CSIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 60-- {text}({text},{text}). Argument indefinite.

Condition Identifier: MLE\$CSQRT_ARG_INDEF Code: ML 60

Description: Note: This message is also associated with FORTRAN condition code FL400.

The argument passed to CSQRT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 61-- {text}({text},{text}). Argument infinite.

Condition Identifier: MLE\$CSQRT_ARG_INF Code: ML 61

Description: Note: This message is also associated with FORTRAN condition code FL401.

The argument passed to CSQRT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 62-- {text}({text},{text}). ABS(argument) + ABS(real part) infinite.

Condition Identifier: MLE\$CSQRT_ARG_RANGE Code: ML 62

Description: Note: This message is also associated with FORTRAN condition code FL402.

The absolute value of the argument passed to CSQRT plus the absolute value of the real part of the argument has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 63-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DACOS_ARG_INDEF Code: ML 63

Description: Note: This message is also associated with FORTRAN condition code FL403.

The argument passed to DACOS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 64-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DACOS_ARG_INF Code: ML 64

Description: Note: This message is also associated with FORTRAN condition code FL404.

The argument passed to DACOS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 65-- {text}({text}). Argument must be in range [-1.0,1.0].

Condition Identifier: MLE\$DACOS_ARG_RANGE Code: ML 65

Description: Note: This message is also associated with FORTRAN condition code FL405.

The argument passed to DACOS must be in the range from -1.0 to 1.0.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DACOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 66-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DASIN_ARG_INDEF Code: ML 66

Description: Note: This message is also associated with FORTRAN condition code FL406.

The argument passed to DASIN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 67-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DASIN_ARG_INF Code: ML 67

Description: Note: This message is also associated with FORTRAN condition code FL407.

The argument passed to DASIN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 68-- {text}({text}). Argument must be in range [-1.0,1.0].

Condition Identifier: MLE\$DASIN_ARG_RANGE Code: ML 68

Description: Note: This message is also associated with FORTRAN condition code FL408.

The argument passed to DASIN must be in the range from -1.0 to 1.0.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DASIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 69-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DATAN_ARG_INDEF Code: ML 69

Description: Note: This message is also associated with FORTRAN condition code FL409.

The argument passed to DATAN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 70-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DATAN2_ARG1_INDEF Code: ML 70

Description: Note: This message is also associated with FORTRAN condition code FL410.

The first argument passed to DATAN2 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 71-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DATAN2_ARG2_INDEF Code: ML 71

Description: Note: This message is also associated with FORTRAN condition code FL411.

The second argument passed to DATAN2 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 72-- {text}(arg1={text},arg2={text}). Arg1 and arg2 may not both be infinite.

Condition Identifier: MLE\$DATAN2_ARGS_INF Code: ML 72

Description: Note: This message is also associated with FORTRAN condition code FL412.

Both the first or second argument passed to DATAN2 are an infinite value, however, one argument must be other than infinite. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite values were generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 73-- {text}(0.0,0.0). One of arg1 or arg2 must be nonzero.

Condition Identifier: MLE\$DATAN2_ARGS_0 Code: ML 73

Description: Note: This message is also associated with FORTRAN condition code FL413.

Either the first or second argument passed to DATAN2 must be nonzero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DATAN2 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 74-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DCOS_ARG_INDEF Code: ML 74

Description: Note: This message is also associated with FORTRAN condition code FL414.

The argument passed to DCOS has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 75-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DCOS_ARG_INF Code: ML 75

Description: Note: This message is also associated with FORTRAN condition code FL415.

The argument passed to DCOS has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 76-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$DCOS_ARG_RANGE Code: ML 76

Description: Note: This message is also associated with FORTRAN condition code FL416.

The absolute value of the argument passed to DCOS must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DCOS routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 77-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DCOSH_ARG_INDEF Code: ML 77

Description: Note: This message is also associated with FORTRAN condition code FL417.

The argument passed to DCOSH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 78-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DCOSH_ARG_INF Code: ML 78

Description: Note: This message is also associated with FORTRAN condition code FL418.

The argument passed to DCOSH has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 79-- {text}({text}). ABS(argument) must be < 4095.*LOG(2).

Condition Identifier: MLE\$DCOSH_ARG_RANGE Code: ML 79

Description: Note: This message is also associated with FORTRAN condition code FL419.

The absolute value of the argument passed to DCOSH must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DCOSH routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 80-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DDIM_ARG1_INDEF Code: ML 80

Description: Note: This message is also associated with FORTRAN condition code FL420.

The first argument passed to DDIM has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 81-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DDIM_ARG2_INDEF Code: ML 81

Description: Note: This message is also associated with FORTRAN condition code FL421.

The second argument passed to DDIM has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 82-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DDIM_ARG1_INF Code: ML 82

Description: Note: This message is also associated with FORTRAN condition code FL422.

The first argument passed to DDIM has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the first argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 83-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DDIM_ARG2_INF Code: ML 83

Description: Note: This message is also associated with FORTRAN condition code FL423.

The second argument passed to DDIM has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 84-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DDIM_RESULT_INF Code: ML 84

Description: Note: This message is also associated with FORTRAN condition code FL424.

The result of the DDIM routine has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 85-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DEXP_ARG_INDEF Code: ML 85

Description: code FL425.

The argument passed to DEXP has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 86-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DEXP_ARG_INF Code: ML 86

Description: Note: This message is also associated with FORTRAN condition code FL426.

The argument passed to DEXP has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 87-- {text}({text}). Argument must be < 4095.*LOG(2).

Condition Identifier: MLE\$DEXP_ARG_TOO_BIG Code: ML 87

Description: Note: This message is also associated with FORTRAN condition code FL427.

The argument passed to DEXP must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DEXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 88-- {text}({text}). Argument must be > -4097.*LOG(2).

Condition Identifier: MLE\$DEXP_ARG_TOO_SMALL Code: ML 88

Description: Note: This message is also associated with FORTRAN condition code FL428.

The argument passed to DEXP must be greater than -4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DEXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 89-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DIM_ARG1_INDEF Code: ML 89

Description: Note: This message is also associated with FORTRAN condition code FL429.

The first argument passed to DIM has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 90-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DIM_ARG2_INDEF Code: ML 90

Description: Note: This message is also associated with FORTRAN condition code FL430.

The second argument passed to DIM has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 91-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DIM_ARG1_INF Code: ML 91

Description: Note: This message is also associated with FORTRAN condition code FL431.

The first argument passed to DIM has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the first argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 92-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DIM_ARG2_INF Code: ML 92

Description: Note: This message is also associated with FORTRAN condition code FL432.

The second argument passed to DIM has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 93-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DIM_RESULT_INF Code: ML 93

Description: Note: This message is also associated with FORTRAN condition code FL433.

The result of the DIM routine has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 94-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DINT_ARG_INDEF Code: ML 94

Description: code FL434.

The argument passed to DINT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 95-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DINT_ARG_INF Code: ML 95

Description: Note: This message is also associated with FORTRAN condition code FL435.

The argument passed to DINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 96-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DLOG_ARG_INDEF Code: ML 96

Description: Note: This message is also associated with FORTRAN condition code FL436.

The argument passed to DLOG has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 97-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DLOG_ARG_INF Code: ML 97

Description: Note: This message is also associated with FORTRAN condition code FL437.

The argument passed to DLOG has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 98-- {text}(0.0). Argument must be > 0.0.

Condition Identifier: MLE\$DLOG_ARG_0 Code: ML 98

Description: Note: This message is also associated with FORTRAN condition code FL438.

The argument passed to DLOG must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DLOG routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 99-- {text}({text}). Argument must be > 0.0.

Condition Identifier: MLE\$DLOG_ARG_NEG Code: ML 99

Description: Note: This message is also associated with FORTRAN condition code FL439.

The argument passed to DLOG must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DLOG routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 100-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DLOG10_ARG_INDEF Code: ML 100

Description: Note: This message is also associated with FORTRAN condition code FL440.

The argument passed to DLOG10 has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 101-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DLOG10_ARG_INF Code: ML 101

Description: Note: This message is also associated with FORTRAN condition code FL441.

The argument passed to DLOG10 has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 102-- {text}(0.0). Argument must be > 0.0.

Condition Identifier: MLE\$DLOG10_ARG_0 Code: ML 102

Description: Note: This message is also associated with FORTRAN condition code FL442.

The argument passed to DLOG10 must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DLOG10 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 103-- {text}({text}). Argument must be > 0.0.

Condition Identifier: MLE\$DLOG10_ARG_NEG Code: ML 103

Description: Note: This message is also associated with FORTRAN condition code FL443.

The argument passed to DLOG10 must be greater than zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DLOG10 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 104-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DMOD_ARG1_INDEF Code: ML 104

Description: Note: This message is also associated with FORTRAN condition code FL444.

The first argument passed to DMOD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 105-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DMOD_ARG2_INDEF Code: ML 105

Description: Note: This message is also associated with FORTRAN condition code FL445.

The second argument passed to DMOD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 106-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DMOD_ARG1_INF Code: ML 106

Description: Note: This message is also associated with FORTRAN condition code FL446.

The first argument passed to DMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the first argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 107-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DMOD_ARG2_INF Code: ML 107

Description: Note: This message is also associated with FORTRAN condition code FL447.

The second argument passed to DMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 108-- {text}(arg1={text},arg2=0.0). Arg2 must be nonzero.

Condition Identifier: MLE\$DMOD_ARG2_0 Code: ML 108

Description: Note: This message is also associated with FORTRAN condition code FL448.

The second argument passed to DMOD must not be zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DMOD routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 109-- {text}(arg1={text},arg2={text}). Arg1/arg2 infinite.

Condition Identifier: MLE\$DMOD_ARGS_RANGE Code: ML 109

Description: Note: This message is also associated with FORTRAN condition code FL449.

The result of dividing the arguments passed to DMOD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 110-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DNINT_ARG_INDEF Code: ML 110

Description: Note: This message is also associated with FORTRAN condition code FL450.

The argument passed to DNINT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 111-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DNINT_ARG_INF Code: ML 111

Description: Note: This message is also associated with FORTRAN condition code FL451.

The argument passed to DNINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 112-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DPROD_ARG1_INDEF Code: ML 112

Description: Note: This message is also associated with FORTRAN condition code FL589.

The first argument passed to DPROD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the first argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 113-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DPROD_ARG2_INDEF Code: ML 113

Description: Note: This message is also associated with FORTRAN condition code FL590.

The second argument passed to DPROD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the second argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 114-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DPROD_ARG1_INF Code: ML 114

Description: Note: This message is also associated with FORTRAN condition code FL591.

The first argument passed to DPROD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the first argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 115-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DPROD_ARG2_INF Code: ML 115

Description: Note: This message is also associated with FORTRAN condition code FL592.

The second argument passed to DPROD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 116-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DPROD_RESULT_INF Code: ML 116

Description: Note: This message is also associated with FORTRAN condition code FL593.

The result of the DPROD routine has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 117-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DSIN_ARG_INDEF Code: ML 117

Description: Note: This message is also associated with FORTRAN condition code FL452.

The argument passed to DSIN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 118-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DSIN_ARG_INF Code: ML 118

Description: Note: This message is also associated with FORTRAN condition code FL453.

The argument passed to DSIN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 119-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$DSIN_ARG_RANGE Code: ML 119

Description: Note: This message is also associated with FORTRAN condition code FL454.

The absolute value of the argument passed to DSIN must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DSIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 120-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DSINH_ARG_INDEF Code: ML 120

Description: Note: This message is also associated with FORTRAN condition code FL455.

The argument passed to DSINH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 121-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DSINH_ARG_INF Code: ML 121

Description: Note: This message is also associated with FORTRAN condition code FL456.

The argument passed to DSINH has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 122-- {text}({text}). ABS(argument) must be < 4095.*LOG(2).

Condition Identifier: MLE\$DSINH_ARG_RANGE Code: ML 122

Description: Note: This message is also associated with FORTRAN condition code FL457.

The absolute value of the argument passed to DSINH must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DSINH routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 123-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DSQRT_ARG_INDEF Code: ML 123

Description: Note: This message is also associated with FORTRAN condition code FL458.

The argument passed to DSQRT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 124-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DSQRT_ARG_INF Code: ML 124

Description: Note: This message is also associated with FORTRAN condition code FL459.

The argument passed to DSQRT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 125-- {text}({text}). Argument must be ≥ 0.0 .

Condition Identifier: MLE\$DSQRT_ARG_RANGE Code: ML 125

Description: Note: This message is also associated with FORTRAN condition code FL460.

The argument passed to DSQRT must be greater than or equal to zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DSQRT routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 126-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DTAN_ARG_INDEF Code: ML 126

Description: Note: This message is also associated with FORTRAN condition code FL461.

The argument passed to DTAN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 127-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$DTAN_ARG_INF Code: ML 127

Description: Note: This message is also associated with FORTRAN condition code FL462.

The argument passed to DTAN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 128-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$DTAN_ARG_RANGE Code: ML 128

Description: Note: This message is also associated with FORTRAN condition code FL463.

The absolute value of the argument passed to DTAN must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DTAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 129-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$DTANH_ARG_INDEF Code: ML 129

Description: Note: This message is also associated with FORTRAN condition code FL464.

The argument passed to DTANH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 130-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DTOD_ARG1_INDEF Code: ML 130

Description: Note: This message is also associated with FORTRAN condition code FL465.

The DTOD routine is called when you raise a double precision base to a double precision power (D1**D2 where D1 and D2 are double precision). The error occurred because the base D1 had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. It causes an error when used in a calculation or passed to a math function. An indefinite value is generally the result of a calculation that cannot be resolved, such as dividing zero by zero, or using an undefined variable.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors causing invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 131-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DTOD_ARG2_INDEF Code: ML 131

Description: Note: This message is also associated with FORTRAN condition code FL466.

The DTOD routine is called when you raise a double precision base to a double precision power (D1**D2 where D1 and D2 are double precision). The error occurred because the exponent D2 had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 132-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DTOD_ARG1_INF Code: ML 132

Description: Note: This message is also associated with FORTRAN condition code FL467.

The DTOD routine is called when you raise a double precision base to a double precision power (DTOD where D1 and D2 are double precision). The error occurred because the base D1 had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or a division by zero.

User Action: Check program logic to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 133-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DTOD_ARG2_INF Code: ML 133

Description: Note: This message is also associated with FORTRAN condition code FL468.

The DTOD routine is called when you raise a double precision value to a double precision power. (D1**D2 where D1 and D2 are double precision). The error occurred because the exponent D2 had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. It causes an error when used in a calculation or passed to a math function. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 134-- {text}(arg1=0.0,arg2={text}). If arg1=0.0, arg2 must be > 0.0.

Condition Identifier: MLE\$DTOD_RESULT_INDEF Code: ML 134

Description: Note: This message is also associated with FORTRAN condition code FL469.

The DTOD routine is called when you raise a double precision base to a double precision power. (D1**D2 where D1 and D2 are double precision.) If the base D1 is zero then the exponent D2 must be greater than zero.

User Action: Be sure that you have specified the correct operands in the expression D1**D2. Then check program logic to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 135-- {text}(arg1={text},arg2={text}). Arg1 must be >= 0.0.

Condition Identifier: MLE\$DTOD_ARG1_NEG Code: ML 135

Description: Note: This message is also associated with FORTRAN condition code FL470.

The first argument passed to DTOD must be greater than or equal to zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the DTOD routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 136-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DTOD_RESULT_INF Code: ML 136

Description: Note: This message is also associated with FORTRAN condition code FL471.

The result of the DTOD routine has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 137-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DTOI_ARG1_INDEF Code: ML 137

Description: Note: This message is also associated with FORTRAN condition code FL472.

The DTOI routine is called when you raise a double precision base to an integer power (D^{**I} where D is double precision and I is integer). The error occurred because the base D had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 138-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DTOI_ARG1_INF Code: ML 138

Description: Note: This message is also associated with FORTRAN condition code FL473.

The DTOI routine is called when you raise a double precision to an integer power (D^{**I} where D is double precision and I is integer). If the value of the base D is zero, the value of the power I must be greater than zero.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: The DTOI routine is called when you raise a double precision value to an integer power (D^{**I} where D is double precision and I is integer). The error occurred because the base D had an infinite value.

Check all previous uses of the operand D to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual

--FATAL ML 139-- {text}(arg1=0.0,arg2={text}). If arg1=0.0, arg2 must be > 0.0.

Condition Identifier: MLE\$DTOI_RESULT_INDEF Code: ML 139

Description: Note: This message is also associated with FORTRAN condition code FL474.

The DTOI routine is called when you raise a double precision value to an integer power (D^{**I} where D is double precision and I is integer). If the value D is zero, then the power I must be greater than zero.

User Action: Be sure than you have specified the correct operands for the expression D^{**I} . Then check program logic to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 140-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DТОI_RESULT_INF Code: ML 140

Description: Note: This message is also associated with FORTRAN condition code FL475.

The DTOI routine is called when you raise a double precision value to an integer power (D^{**I} where D is double precision and I is integer). This operation generated an infinite result.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. The probable cause is a value for D or I that is infinite or that approached the limits of the computer.

User Action: Check both operands of the exponential expression for incorrect values. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual

--FATAL ML 141-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$DТОX_ARG1_INDEF Code: ML 141

Description: Note: This message is also associated with FORTRAN condition code FL476.

The routine DTOX is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The error occurred because the exponent X had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand D of the expression D^{**X} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 142-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$DТОX_ARG2_INDEF Code: ML 142

Description: Note: This message is also associated with FORTRAN condition code FL477.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The error occurred because the exponent X had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 143-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$DТОX_ARG1_INF Code: ML 143

Description: Note: This message is also associated with FORTRAN condition code FL478.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The error occurred because the operand D had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand D of D^{**X} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 144-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$DTOX_ARG2_INF Code: ML 144

Description: Note: This message is also associated with FORTRAN condition code FL479.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The error occurred because the exponent X had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 145-- {text}(arg1=0.0,arg2={text}). If arg1=0.0, arg2 must be > 0.0.

Condition Identifier: MLE\$DTOX_RESULT_INDEF Code: ML 145

Description: Note: This message is also associated with FORTRAN condition code FL480.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). If the value D is zero then the exponent X must be greater than zero.

User Action: Check the exponent X of the expression D^{**X} for a zero value.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 146-- {text}(arg1={text},arg2={text}). Arg1 must be >= 0.0.

Condition Identifier: MLE\$DTOX_ARG1_NEG Code: ML 146

Description: Note: This message is also associated with FORTRAN condition code FL481.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The value of the exponent X must be greater than or equal to zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 147-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$DTOX_RESULT_INF Code: ML 147

Description: Note: This message is also associated with FORTRAN condition code FL482.

The DTOX routine is called when you raise a double precision value to a real power (D^{**X} where D is double precision and X is real). The error occurred because the operation generated an infinite result. The probable cause is that one or both of the operands D and X contain incorrect values.

User Action: Check the operands D and X of D^{**X} for incorrect values. Possible causes of incorrect values are improperly defined variables, bad input data, or improper data type (a real variable should contain a floating-point number, etc.)

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 148-- {text}({text},{text}). Arg1 indefinite.

Condition Identifier: MLE\$DTOZ_ARG1_INDEF Code: ML 148

Description: Note: This message is also associated with FORTRAN condition code FL483.

The DTOZ routine is called when you raise a double precision value to a complex power (D^{**Z} where D is double precision and Z is complex). The error occurred because the operand D had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand D to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 149-- {text}({text},{text},{text}). Arg2 indefinite.

Condition Identifier: MLE\$DTOZ_ARG2_INDEF Code: ML 149

Description: Note: This message is also associated with FORTRAN condition code FL484.

The DTOZ routine is called when you raise a double precision value to a complex power (D^{**Z} where D is double precision and Z is complex). The error occurred because the exponent Z had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 150-- {text}({text},{text},{text}). Arg1 infinite.

Condition Identifier: MLE\$DTOZ_ARG1_INF Code: ML 150

Description: Note: This message is also associated with FORTRAN condition code FL485.

The DTOZ routine is called when you raise a double precision value to a complex power (D^{**Z} where D is double precision and Z is complex). The error occurred because the operand D had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand D of D^{**Z} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 151-- {text}({text},{text},{text}). Arg2 infinite.

Condition Identifier: MLE\$DTOZ_ARG2_INF Code: ML 151

Description: Note: This message is also associated with FORTRAN condition code FL486.

The DTOZ routine is called when you raise a double precision value to a complex power (D^{**Z} where D is double precision and Z is complex). The error occurred because the exponent Z had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual. Refer also to the description of complex constants for the correct format of a complex value.

--FATAL ML 152-- {text}(0.0,({text},{text})). RE(Arg2) must be > 0 and IM(Arg2)=0.

Condition Identifier: MLE\$DTOZ_RESULT_INDEF Code: ML 152

Description: Note: This message is also associated with FORTRAN condition code FL487.

The DTOZ routine is called when you raise a double precision value to a complex power (D^{**Z} where D is double precision and Z is complex). If the operand D is zero, the exponent X must also be zero.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Be sure that you have specified the correct operands in the expression DF^{**X} . Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 154-- {text}({text},{text},{text}). Result infinite.

Condition Identifier: MLE\$DZTOZ_RESULT_INF Code: ML 154

Description: Note: This message is also associated with FORTRAN condition code FL489.

The DZTOZ routine is called when you raise a double precision value to a complex power (D**Z where D is double precision and Z is complex). The error occurred because the exponentiation operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands D and Z of the expression D**Z for incorrect values. Possible causes of incorrect values are improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 155-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ERF_ARG_INDEF Code: ML 155

Description: Note: This message is also associated with FORTRAN condition code FL490.

The argument passed to ERF has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 156-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$ERFC_ARG_INDEF Code: ML 156

Description: Note: This message is also associated with FORTRAN condition code FL491.

The argument passed to ERFC has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation, or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 157-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$EXP_ARG_INDEF Code: ML 157

Description: Note: This message is also associated with FORTRAN condition code FL492.

The argument passed to EXP has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 158-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$EXP_ARG_INF Code: ML 158

Description: Note: This message is also associated with FORTRAN condition code FL493.

The argument passed to EXP has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 159-- {text}({text}). Argument must be < 4095.*LOG(2).

Condition Identifier: MLE\$EXP_ARG_TOO_BIG Code: ML 159

Description: Note: This message is also associated with FORTRAN condition code FL494.

The argument passed to EXP must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the EXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 160-- {text}({text}). Argument must be > -4097.*LOG(2).

Condition Identifier: MLE\$EXP_ARG_TOO_SMALL Code: ML 160

Description: Note: This message is also associated with FORTRAN condition code FL495.

The argument passed to EXP must be greater than -4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the EXP routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 161-- {text}(arg1={text},arg2={text}). Arithmetic overflow.

Condition Identifier: MLE\$IDIM_RESULT_INF Code: ML 161

Description: Note: This message is also associated with FORTRAN condition code FL496.

The IDIM function generated a result that exceeded the largest integer that can be represented in internal binary format.

User Action: Check the arguments for incorrect values. Possible cause of incorrect values are improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 162-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$IDNINT_ARG_INDEF Code: ML 162

Description: Note: This message is also associated with FORTRAN condition code FL497.

The argument passed to IDNINT had an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 163-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$IDNINT_ARG_INF Code: ML 163

Description: Note: This message is also associated with FORTRAN condition code FL498.

The argument passed to IDNINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 164-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$ITOD_ARG2_INDEF Code: ML 164

Description: Note: This message is also associated with FORTRAN condition code FL499.

The ITOD routine is called when you raise an integer value to a double precision power (I^{**D} where I is integer and D is double precision). The error occurred because the exponent D had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 165-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$ITOD_ARG2_INF Code: ML 165

Description: Note: This message is also associated with FORTRAN condition code FL500.

The ITOD routine is called when you raise an integer value to a double precision power (I^{**D} where I is integer and D is double precision). The error occurred because the exponent D had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent in the expression I^{**D} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 166-- {text}(arg1=0,arg2={text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$ITOD_RESULT_INDEF Code: ML 166

Description: Note: This message is also associated with FORTRAN condition code FL501.

The routine ITOD is called when you raise an integer value to a double precision power (I^{**D} where I is integer and D is double precision). The error occurred because the exponent D had a value less than or equal to zero.

User Action: Check all previous uses of the exponent of the expression I^{**D} to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 167-- {text}(arg1={text},arg2={text}). Arg1 must be ≥ 0.0 .

Condition Identifier: MLE\$ITOD_ARG1_NEG Code: ML 167

Description: Note: This message is also associated with FORTRAN condition code FL502.

The ITOD routine is called when you raise an integer value to a double precision power (I^{**D} where I is integer and D is double precision). The error occurred because the value of I was less than zero.

User Action: Be sure that you have specified the correct operand I in the expression I^{**D} . Then check all previous uses of the operand to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 168-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$ITOD_RESULT_INF Code: ML 168

Description: Note: This message is also associated with FORTRAN condition code FL503.

The ITOD routine is called when you raise an integer value to a double precision power (I^{**D} where I is integer and D is double precision). The error occurred because this operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operands to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 169-- {text}(arg1={text},arg2={text}). Arithmetic overflow.

Condition Identifier: MLE\$ITOI_RESULT_INF Code: ML 169

Description: Note: This message is also associated with FORTRAN condition code FL504.

The ITOI routine is called when you raise an integer value to an integer power ($I1^{**I2}$ where I1 and I2 are integer). The error occurred because this operation produced an arithmetic overflow.

An arithmetic overflow occurs when a computation produces a value whose magnitude exceeds the largest value that can be represented in internal binary format. The probable cause of the error is that one or both of the operands I1 and I2 contain in the expression $I1^{**I2}$ contain incorrect values.

User Action: Be sure that you have specified the correct operands in the exponential expression. Then check all previous uses of the operands to determine the cause of the incorrect value.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 170-- {text}(arg1=0,arg2={text}). Arg2 must be > 0.0 .

Condition Identifier: MLE\$ITOI_RESULT_INDEF Code: ML 170

Description: Note: This message is also associated with FORTRAN condition code FL505.

The ITOI routine is called when you raise an integer value to an integer power ($I1^{**I2}$ where I1 and I2 are integer). If the operand I1 is zero, then the operand I2 must be greater than zero.

User Action: Be sure that you have specified the correct operands in the exponential expression. Then check program logic to determine how the incorrect value(s) was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 171-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$ITOX_ARG2_INDEF Code: ML 171

Description: Note: This message is also associated with FORTRAN condition code FL506.

The ITOX routine is called when you raise an integer value to a real power (I^{**X} where I is integer and X is real). The error occurred because the exponent X had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated.
 Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 172-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$ITOX_ARG2_INF Code: ML 172

Description: Note: This message is also associated with FORTRAN condition code FL507.

The ITOX routine is called when you raise an integer value to a real power (I^{**X} where I is integer and X is real). The error occurred because the exponent X had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent to determine how the infinite value was generated.
 Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 173-- {text}(arg1=0,arg2={text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$ITOX_RESULT_INDEF Code: ML 173

Description: Note: This message is also associated with FORTRAN condition code FL508.

The ITOX routine is called when you raise an integer value to a real power (I^{**X} where I is integer and X is real). If the operand I is zero, then the exponent X must be greater than zero.

User Action: Be sure that you have specified the correct operands in the expression I^{**X} . Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 174-- {text}(arg1={text},arg2={text}). Arg1 must be >= 0.0.

Condition Identifier: MLE\$ITOX_ARG1_NEG Code: ML 174

Description: Note: This message is also associated with FORTRAN condition code FL509.

The ITOX routine is called when you raise an integer value to a real power (I^{**X} where I is integer and X is real). The operand I must be greater than or equal to zero.

User Action: Be sure that you have specified the correct operand I in I^{**X} . Then check all previous uses of the operand to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 175-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$ITOX_RESULT_INF Code: ML 175

Description: Note: This message is also associated with FORTRAN condition code FL510.

The ITOX routine is called when you raise an integer value to a real power (I^{**X} where I is integer and X is real). The error occurred because the operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands for incorrect values. Possible causes of incorrect values are improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 176-- {text}({text},{text},{text}). Arg2 indefinite.

Condition Identifier: MLE\$IT0Z_ARG2_INDEF Code: ML 176

Description: Note: This message is also associated with FORTRAN condition code FL511.

The ITOZ routine is called when you raise an integer value to a complex power (I^{**X} where I is integer and Z is complex). The error occurred because the exponent Z had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 177-- {text}({text},{text},{text}). Arg2 infinite.

Condition Identifier: MLE\$IT0Z_ARG2_INF Code: ML 177

Description: Note: This message is also associated with FORTRAN condition code FL512.

The ITOZ routine is called when you raise an integer value to a complex power (I^{**Z} where I is integer and Z is complex). The error occurred because the exponent Z had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Possible errors include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 178-- {text}(0,({text},{text})). RE(Arg2) must be > 0 and IM(Arg2)=0.

Condition Identifier: MLE\$IT0Z_RESULT_INDEF Code: ML 178

Description: Note: This message is also associated with FORTRAN condition code FL513.

The ITOZ routine is called when you raise an integer value to a complex power (I^{**Z} where I is integer and Z is complex). If the operand I is zero, then the exponent Z must be greater than zero.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Be sure that you have specified the correct operands in the expression I^{**Z} . Then check all previous uses of the arguments to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 179-- {text}({text},{text},{text}). Result infinite.

Condition Identifier: MLE\$IT0Z_RESULT_INF Code: ML 179

Description: Note: This message is also associated with FORTRAN condition code FL514.

The ITOZ routine is called when you raise an integer value to a complex power (I^{**Z} where I is integer and Z is complex). The error occurred because the operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands for incorrect values. Possible causes of infinite values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 181-- {text}(arg1={text},arg2=0). Arg2 must be nonzero.

Condition Identifier: MLE\$MOD_ARG2_0 Code: ML 181

Description: Note: This message is also associated with FORTRAN condition code FL515.

The second argument passed to MOD must not be zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the MOD function in the FORTRAN Language Definition Usage Manual.

--FATAL ML 182-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$NINT_ARG_INDEF Code: ML 182

Description: Note: This message is also associated with FORTRAN condition code FL516.

The argument passed to NINT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 183-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$NINT_ARG_INF Code: ML 183

Description: Note: This message is also associated with FORTRAN condition code FL517.

The argument passed to NINT has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 184-- {text}({text}). Argument must be ≤ 53.0374219959898 .

Condition Identifier: MLE\$ERFC_ARG_RANGE Code: ML 184

Description: Note: This message is also associated with FORTRAN condition code FL594.

The argument passed to ERFC must be less than or equal to 53.037.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the ERFC routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 185-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$SIN_ARG_INDEF Code: ML 185

Description: Note: This message is also associated with FORTRAN condition code FL522.

The argument passed to SIN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 186-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$SIN_ARG_INF Code: ML 186

Description: Note: This message is also associated with FORTRAN condition code FL523.

The argument passed to SIN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 187-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$SIN_ARG_RANGE Code: ML 187

Description: Note: This message is also associated with FORTRAN condition code FL524.

The absolute value of the argument passed to SIN must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the SIN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 188-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$SINH_ARG_INDEF Code: ML 188

Description: Note: This message is also associated with FORTRAN condition code FL525.

The argument passed to SINH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 189-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$SINH_ARG_INF Code: ML 189

Description: Note: This message is also associated with FORTRAN condition code FL526.

The argument passed to SINH has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 190-- {text}({text}). ABS(argument) must be < 4095.*LOG(2).

Condition Identifier: MLE\$SINH_ARG_RANGE Code: ML 190

Description: Note: This message is also associated with FORTRAN condition code FL527.

The absolute value of the argument passed to SINH must be less than 4095.*LOG(2). (4095.*LOG(2) is approximately 2838.4377.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the SINH routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 191-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$SQRT_ARG_INDEF Code: ML 191

Description: Note: This message is also associated with FORTRAN condition code FL528.

The argument passed to SQRT has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 192-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$SQRT_ARG_INF Code: ML 192

Description: Note: This message is also associated with FORTRAN condition code FL529.

The argument passed to SQRT has an infinite value. An infinite value is a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 193-- {text}({text}). Argument must be ≥ 0.0 .

Condition Identifier: MLE\$SQRT_ARG_NEG Code: ML 193

Description: Note: This message is also associated with FORTRAN condition code FL530.

The argument passed to SQRT must be greater than or equal to zero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the SQRT routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 194-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$TAN_ARG_INDEF Code: ML 194

Description: Note: This message is also associated with FORTRAN condition code FL531.

The argument passed to TAN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 195-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$TAN_ARG_INF Code: ML 195

Description: Note: This message is also associated with FORTRAN condition code FL532.

The argument passed to TAN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 196-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$TAN_ARG_RANGE Code: ML 196

Description: Note: This message is also associated with FORTRAN condition code FL533.

The absolute value of the argument passed to TAN must be less than 2.**47. (2.**47 is approximately 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the TAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 197-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$TANH_ARG_INDEF Code: ML 197

Description: Note: This message is also associated with FORTRAN condition code FL534.

The argument passed to TANH has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 198-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$XTOD_ARG1_INDEF Code: ML 198

Description: Note: This message is also associated with FORTRAN condition code FL535.

The XTOD routine is called when you raise a real value to a double precision power (X**D where X is real and D is double precision). The error occurred because the operand X had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand X of X**D to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 199-- {text}(arg1={text},arg2={text}). Arg2 indefinite.

Condition Identifier: MLE\$XTOD_ARG2_INDEF Code: ML 199

Description: Note: This message is also associated with FORTRAN condition code FL536.

The XTOD routine is called when you raise a real value to a double precision power (X**D where X is real and D is double precision). The error occurred because the exponent D had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 200-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$XTOD_ARG1_INF Code: ML 200

Description: Note: This message is also associated with FORTRAN condition code FL537.

The XTOD routine is called when you raise a real value to a double precision power (X^{**D} where X is real and D is double precision). The error occurred because the operand X had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand X of X^{**D} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 201-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$XTOD_ARG2_INF Code: ML 201

Description: code FL538.

The XTOD routine is called when you raise a real value to a double precision power (X^{**D} where X is real and D is double precision). The error occurred because the exponent D had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand X of X^{**D} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 202-- {text}(arg1=0.0,arg2={text}). Arg2 must be ≥ 0.0 .

Condition Identifier: MLE\$XTOD_RESULT_INDEF Code: ML 202

Description: Note: This message is also associated with FORTRAN condition code FL539.

The XTOD routine is called when you raise a real value to a double precision power (X^{**D} where X is real and D is double precision). If the operand X is zero, the exponent D must be greater than zero.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Be sure that you have specified the correct operands in the expression X^{**D} . Then check program logic to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 203-- {text}(arg1={text},arg2={text}). Arg1 must be ≥ 0.0 .

Condition Identifier: MLE\$XTOD_ARG1_NEG Code: ML 203

Description: code FL540.

The XTOD routine is called when you raise a real value to a double precision power (X^{**D} where X is real and D is double precision). The exponent D must be greater than or equal to zero.

User Action: Be sure that you have specified the correct operands for the expression X^{**D} . Then check program logic to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 204-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$XTOD_RESULT_INF Code: ML 204

Description: Note: This message is also associated with FORTRAN condition code FL541.

The XTOD routine is called when you raise a real value to a double precision power (X^{**D} where X is real and D is double precision). The error occurred because the operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 205-- {text}(arg1={text},arg2={text}). Arg1 indefinite.

Condition Identifier: MLE\$XTOI_ARG1_INDEF Code: ML 205

Description: Note: This message is also associated with FORTRAN condition code FL542.

The XTOI routine is called when you raise a real value to a real power ($X1^{**X2}$ where X1 and X2 are real). The error occurred because the operand X1 had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand X of X^{**I} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 206-- {text}(arg1={text},arg2={text}). Arg1 infinite.

Condition Identifier: MLE\$XTOI_ARG1_INF Code: ML 206

Description: Note: This message is also associated with FORTRAN condition code FL543.

The XTOI routine is called when you raise a real value to a real power ($X1^{**X2}$ where X1 and X2 are real). The error occurred because the operand X had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Possible errors include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 207-- {text}(arg1=0.0,arg2={text}). Arg2 must be > 0.

Condition Identifier: MLE\$XTOI_RESULT_INDEF Code: ML 207

Description: Note: This message is also associated with FORTRAN condition code FL554.

The XTOI routine is called when you raise a real value to an integer power (X^{**I} where X is real and I is integer). If the operand X is zero, then the exponent I must be greater than zero.

User Action: Be sure that you have specified the correct operands in the expression X^{**I} . Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 208-- {text}(arg1 = {text},arg2 = {text}). Result infinite.

Condition Identifier: MLE\$XTOI_RESULT_INF Code: ML 208

Description: Note: This message is also associated with FORTRAN condition code FL545.

The XTOI routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the operation produced an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands of $X^{**}I$ for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, bad input data, or logic errors that lead to invalid calculations.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 209-- {text}(arg1 = {text},arg2 = {text}). Arg1 indefinite.

Condition Identifier: MLE\$XTOX_ARG1_INDEF Code: ML 209

Description: Note: This message is also associated with FORTRAN condition code FL546.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the operand $X1$ had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors causing invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 210-- {text}(arg1 = {text},arg2 = {text}). Arg2 indefinite.

Condition Identifier: MLE\$XTOX_ARG2_INDEF Code: ML 210

Description: Note: This message is also associated with FORTRAN condition code FL547.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the exponent $X2$ had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors causing invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 211-- {text}(arg1 = {text},arg2 = {text}). Arg1 infinite.

Condition Identifier: MLE\$XTOX_ARG1_INF Code: ML 211

Description: Note: This message is also associated with FORTRAN condition code FL548.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the operand $X1$ had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 212-- {text}(arg1={text},arg2={text}). Arg2 infinite.

Condition Identifier: MLE\$XTOX_ARG2_INF Code: ML 212

Description: Note: This message is also associated with FORTRAN condition code FL549.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the exponent $X2$ had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result. Possible causes of infinite values include operations that involve very small or very large numbers, and division by zero.

User Action: Check program logic to determine how the infinite value was generated. Possible errors include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 213-- {text}(arg1=0.0,arg2={text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$XTOX_RESULT_INDEF Code: ML 213

Description: Note: This message is also associated with FORTRAN condition code FL550.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). If the operand $X1$ is zero, then the exponent $X2$ must be greater than zero.

User Action: Be sure that you have specified the correct arguments in the expression $X^{**}X$. Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 214-- {text}(arg1={text},arg2={text}). Arg1 must be >= 0.0.

Condition Identifier: MLE\$XTOX_ARG1_NEG Code: ML 214

Description: Note: This message is also associated with FORTRAN condition code FL551.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The exponent $X2$ must be greater than or equal to zero.

User Action: Be sure that you have specified the correct operands in the expression $X1^{**}X2$. Then check program logic to determine how the incorrect value was generated.

Further Information: Refer to the description of exponentiation in the FORTRAN Language Definition Usage Manual.

--FATAL ML 215-- {text}(arg1={text},arg2={text}). Result infinite.

Condition Identifier: MLE\$XTOX_RESULT_INF Code: ML 215

Description: Note: This message is also associated with FORTRAN condition code FL552.

The XTOX routine is called when you raise a real value to a real power ($X1^{**}X2$ where $X1$ and $X2$ are real). The error occurred because the operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. The probable cause of the error is that one or both of the operands $X1$ and $X2$ in the expression $X1^{**}X2$ contained an incorrect value or a value whose magnitude approached the limits of the computer.

User Action: Check the operands for incorrect values. Possible errors include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 216-- {text}({text},({text},{text})). Arg1 indefinite.

Condition Identifier: MLE\$XTOZ_ARG1_INDEF Code: ML 216

Description: Note: This message is also associated with FORTRAN condition code FL553.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). The error occurred because the operand X had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generally the result of a calculation that cannot be resolved, such as dividing zero by zero, or use of an undefined variable.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 217-- {text}({text},({text},{text})). Arg2 indefinite.

Condition Identifier: MLE\$XTOZ_ARG2_INDEF Code: ML 217

Description: Note: This message is also associated with FORTRAN condition code FL554.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). The error occurred because the exponent Z had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors causing invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 218-- {text}({text},({text},{text})). Arg1 infinite.

Condition Identifier: MLE\$XTOZ_ARG1_INF Code: ML 218

Description: Note: This message is also associated with FORTRAN condition code FL555.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). The error occurred because the operand X had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 219-- {text}({text},({text},{text})). Arg2 infinite.

Condition Identifier: MLE\$XTOZ_ARG2_INF Code: ML 219

Description: Note: This message is also associated with FORTRAN condition code FL556.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). The error occurred because the exponent Z had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Possible causes include undefined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a division by zero.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 220-- {text}(0.0,({text},{text})). RE(Arg2) must be > 0 and IM(Arg2)=0.

Condition Identifier: MLE\$XTOZ_RESULT_INDEF Code: ML 220

Description: Note: This message is also associated with FORTRAN condition code FL557.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). If the operand X is zero, then the real part of the exponent Z must be greater than zero and the imaginary part must be equal to zero.

User Action: Be sure that you have specified the correct operands in the expression X^{**Z} . Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 221-- {text}({text},({text},{text})). Result infinite.

Condition Identifier: MLE\$XTOZ_RESULT_INF Code: ML 221

Description: Note: This message is also associated with FORTRAN condition code FL558.

The routine XTOZ is called when you raise a real value to a complex power (X^{**Z} where X is real and Z is complex). The error occurred because the operation generated an infinite result.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. The probable cause of the error is that one or both of the operands in the expression X^{**Z} contained an incorrect value or a value whose magnitude approached the limits of the computer.

User Action: Check the operands for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 222-- {text}(((text),{text})),{text}). Arg1 indefinite.

Condition Identifier: MLE\$ZTOD_ARG1_INDEF Code: ML 222

Description: Note: This message is also associated with FORTRAN condition code FL559.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). The error occurred because the operand Z had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 223-- {text}(((text),{text})),{text}). Arg2 indefinite.

Condition Identifier: MLE\$ZTOD_ARG2_INDEF Code: ML 223

Description: Note: This message is also associated with FORTRAN condition code FL560.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). The error occurred because the exponent D had an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check program logic to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 224-- {text}({text},{text}},{text}). Arg1 infinite.

Condition Identifier: MLE\$ZTOD_ARG1_INF Code: ML 224

Description: code FL561.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). The error occurred because the operand Z had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check program logic to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a division by zero.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 225-- {text}({text},{text}},{text}). Arg2 infinite.

Condition Identifier: MLE\$ZTOD_ARG2_INF Code: ML 225

Description: Note: This message is also associated with FORTRAN condition code FL562.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). The error occurred because the exponent D had an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the second argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 226-- {text}(0.0,{text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$ZTOD_RESULT_INDEF Code: ML 226

Description: Note: This message is also associated with FORTRAN condition code FL563.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). If the operand Z is zero, then the exponent D must be greater than zero.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent D of the expression Z^{**D} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 227-- {text}({text},{text}},{text}). Result infinite.

Condition Identifier: MLE\$ZTOD_RESULT_INF Code: ML 227

Description: Note: This message is also associated with FORTRAN condition code FL564.

The ZTOD routine is called when you raise a complex value to a double precision power (Z^{**D} where Z is complex and D is double precision). The error occurred because the operation generated an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands Z and D of D^{**Z} for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 228-- {text}({text},{text}),{text}). Arg1 indefinite.

Condition Identifier: MLE\$ZTOI_ARG1_INDEF Code: ML 228

Description: Note: This message is also associated with FORTRAN condition code FL565.

The ZTOI routine is called when you raise a complex value to an integer power (Z^{**I} where Z is complex and I is integer). The error occurred because the operand Z contained an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand Z of Z^{**I} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 229-- {text}({text},{text}),{text}). Arg1 infinite.

Condition Identifier: MLE\$ZTOI_ARG1_INF Code: ML 229

Description: Note: This message is also associated with FORTRAN condition code FL566.

The ZTOI routine is called when you raise a complex value to an integer power (Z^{**I} where Z is complex and I is integer). The error occurred because the operand Z contained an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand Z of Z^{**I} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 230-- {text}({text},{text}),{text}). Result infinite.

Condition Identifier: MLE\$ZTOI_RESULT_INF Code: ML 230

Description: Note: This message is also associated with FORTRAN condition code FL567.

The ZTOI routine is called when you raise a complex value to an integer power (Z^{**I} where Z is complex and I is integer). The error occurred because the operation generated an infinite result.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands of the expression Z^{**I} for incorrect values. Possible causes of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, and bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 231-- {text}(0.0,{text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$ZTOI_RESULT_INDEF Code: ML 231

Description: Note: This message is also associated with FORTRAN condition code FL568.

The ZTOI routine is called when you raise a complex value to an integer power (Z^{**I} where Z is complex and I is integer). If the operand Z is zero, then the exponent I must be greater than zero.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Be sure that you have specified the correct operands in the expression Z^{**I} . Then check all previous uses of the operands to determine how the incorrect value(s) was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 232-- {text}({text},{text},{text}). Arg1 indefinite.

Condition Identifier: MLE\$ZTOX_ARG1_INDEF Code: ML 232

Description: Note: This message is also associated with FORTRAN condition code FL569.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). The error occurred because the operand Z contained an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand Z of Z^{**X} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 233-- {text}({text},{text},{text}). Arg2 indefinite.

Condition Identifier: MLE\$ZTOX_ARG2_INDEF Code: ML 233

Description: Note: This message is also associated with FORTRAN condition code FL570.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). The error occurred because the exponent X contained an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent X of Z^{**X} to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 234-- {text}({text},{text},{text}). Arg1 infinite.

Condition Identifier: MLE\$ZTOX_ARG1_INF Code: ML 234

Description: Note: This message is also associated with FORTRAN condition code FL571.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). The error occurred because the operand Z contained an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand Z of Z^{**X} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 235-- {text}({text},{text},{text}). Arg2 infinite.

Condition Identifier: MLE\$ZTOX_ARG2_INF Code: ML 235

Description: Note: This message is also associated with FORTRAN condition code FL572.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). The error occurred because the exponent X contained an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent X of Z^{**X} to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 236-- {text}(0.0,{text}). Arg2 must be > 0.0.

Condition Identifier: MLE\$ZTOX_RESULT_INDEF Code: ML 236

Description: Note: This message is also associated with FORTRAN condition code FL573.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). If the operand Z is zero, then the exponent X must be greater than zero.

User Action: Be sure that you have specified the correct operands in the expression Z^{**X} . Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 237-- {text}({text},{text}),{text}). Result infinite.

Condition Identifier: MLE\$ZTOX_RESULT_INF Code: ML 237

Description: Note: This message is also associated with FORTRAN condition code FL574.

The ZTOX routine is called when you raise a complex value to a real power (Z^{**X} where Z is complex and X is real). The error occurred because the operation generated an infinite result.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check the operands of the expression Z^{**X} for incorrect values. Possible cause of incorrect values include improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 238-- {text}({text},{text}),({text},{text}). Arg1 indefinite.

Condition Identifier: MLE\$ZTOZ_ARG1_INDEF Code: ML 238

Description: Note: This message is also associated with FORTRAN condition code FL575.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**Z2}$ where $Z1$ and $Z2$ are complex). The error occurred because the operand $Z1$ contained an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the operand $Z1$ of $Z1^{**Z2}$ to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 239-- {text}({text},{text}),({text},{text}). Arg2 indefinite.

Condition Identifier: MLE\$ZTOZ_ARG2_INDEF Code: ML 239

Description: Note: This message is also associated with FORTRAN condition code FL576.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**Z2}$ where $Z1$ and $Z2$ are complex). The error occurred because the exponent $Z2$ contained an indefinite value.

An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the exponent $Z2$ of $Z1^{**Z2}$ to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 240-- {text}({text},{text}),({text},{text}). Arg1 infinite.

Condition Identifier: MLE\$ZTOZ_ARG1_INF Code: ML 240

Description: Note: This message is also associated with FORTRAN condition code FL577.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**}Z2$ where $Z1$ and $Z2$ are complex). The error occurred because the operand $Z1$ contained an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operand $Z1$ of $Z1^{**}Z2$ to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 241-- {text}({text},{text}),({text},{text}). Arg2 infinite.

Condition Identifier: MLE\$ZTOZ_ARG2_INF Code: ML 241

Description: Note: This message is also associated with FORTRAN condition code FL578.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**}Z2$ where $Z1$ and $Z2$ are complex). The error occurred because the exponent $Z2$ contained an infinite value.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the exponent $Z2$ of $Z1^{**}Z2$ to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 242-- {text}(0.0,({text},{text})). Arg1 must be nonzero.

Condition Identifier: MLE\$ZTOZ_RESULT_INDEF Code: ML 242

Description: Note: This message is also associated with FORTRAN condition code FL579.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**}Z2$ where $Z1$ and $Z2$ are complex). If the operand $Z1$ is zero, the exponent $Z2$ must not also be zero.

User Action: Be sure that you have correct operands in the expression $Z1^{**}Z2$. Then check all previous uses of the operands to determine how the incorrect value was generated.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 243-- {text}({text},{text}),({text},{text}). Result infinite.

Condition Identifier: MLE\$ZTOZ_RESULT_INF Code: ML 243

Description: Note: This message is also associated with FORTRAN condition code FL580.

The ZTOZ routine is called when you raise a complex value to a complex power ($Z1^{**}Z2$ where $Z1$ and $Z2$ are complex). The error occurred because the operation generated an infinite result.

An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the operands of the expression $Z1^{**}Z2$ to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 244-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$SIND_ARG_INDEF Code: ML 244

Description: Note: This message is also associated with FORTRAN condition code FL596.

The argument passed to SIND has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 245-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$SIND_ARG_INF Code: ML 245

Description: Note: This message is also associated with FORTRAN condition code FL597.

The argument passed to SIND has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 246-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$SIND_ARG_RANGE Code: ML 246

Description: Note: This message is also associated with FORTRAN condition code FL598.

The absolute value of the argument passed to SIND must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the SIND routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 247-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$COSD_ARG_INDEF Code: ML 247

Description: Note: This message is also associated with FORTRAN condition code FL599.

The argument passed to COSD has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated.

Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 248-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$COSD_ARG_INF Code: ML 248

Description: Note: This message is also associated with FORTRAN condition code FL600.

The argument passed to COSD has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated.

Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 249-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$COSD_ARG_RANGE Code: ML 249

Description: Note: This message is also associated with FORTRAN condition code FL601.

The absolute value of the argument passed to COSD must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the COSD routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 250-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$TAND_ARG_INDEF Code: ML 250

Description: Note: This message is also associated with FORTRAN condition code FL602.

The argument passed to TAND has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing zero by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 251-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$TAND_ARG_INF Code: ML 251

Description: Note: This message is also associated with FORTRAN condition code FL603.

The argument passed to TAND has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of expressions in the FORTRAN Language Definition Usage Manual.

--FATAL ML 252-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$TAND_ARG_RANGE Code: ML 252

Description: Note: This message is also associated with FORTRAN condition code FL604.

The absolute value of the argument passed to TAND must be less than 2.**47. (2.**47 = 1.4074E+14.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the TAND routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 253-- {text}({text}). Argument must not be an exact odd multiple of 90.0.

Condition Identifier: MLE\$TAND_RESULT_INF Code: ML 253

Description: Note: This message is also associated with FORTRAN condition code FL605.

The argument to TAND must not be an exact odd multiple of 90.0. (The tangent of an angle that is an exact odd multiple of 90 is mathematically undefined.)

User Action: Be sure that you have specified the correct argument and that the argument is not an exact odd multiple of 90. Be sure that the argument is in degrees.

Further Information: Refer to the description of the TAND function in the FORTRAN Language Definition Usage Manual.

--FATAL ML 254-- {text}({text}). Argument indefinite.

Condition Identifier: MLE\$COTAN_ARG_INDEF Code: ML 254

Description: This message is also associated with FORTRAN condition code FL614.

The argument passed to COTAN has an indefinite value. An indefinite value has an internal representation that does not correspond to a number. An indefinite value is generated either when an undefined variable is used in a calculation or when a calculation cannot be resolved, such as dividing by zero.

User Action: Check all previous uses of the argument to determine how the indefinite value was generated. Check for logic errors that result in invalid calculations, improperly defined variables, or bad input data.

Further Information: Refer to the description of the COTAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 255-- {text}({text}). Argument infinite.

Condition Identifier: MLE\$COTAN_ARG_INF Code: ML 255

Description: This message is also associated with FORTRAN condition code FL615.

The argument passed to COTAN has an infinite value. An infinite value represents a number that is larger than the largest number that can be represented in floating-point format. An infinite value is generated when a calculation produces a mathematically infinite result such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of the argument to determine how the infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of the COTAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 256-- {text}({text}). ABS(argument) must be < 2.47.**

Condition Identifier: MLE\$COTAN_ARG_RANGE Code: ML 256

Description: This message is also associated with FORTRAN condition code FL616

The absolute value of the argument passed to COTAN must be less than 2.**47; (2.**47 = 1.4074E+14).

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the COTAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 257-- {text}(arg2={text},arg3={text}). Starting bit must be >= 0.

Condition Identifier: MLE\$EXTB_ARG2_NEG Code: ML 257

Description: Note: This message is also associated with FORTRAN condition code FL617.

The second argument passed to EXTB, which specifies the first bit to be extracted, has a negative value. The argument must be an integer in the range 0 - 63.

User Action: Be sure that you have specified the correct argument and that the argument is an integer in the range 0 - 63 inclusive.

Further Information: Refer to the description of the EXTB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 258-- {text}(arg2={text},arg3={text}). Length must be >= 0.

Condition Identifier: MLE\$EXTB_ARG3_NEG Code: ML 258

Description: Note: This message is also associated with FORTRAN condition code FL618.

The third argument passed to EXTB, which specifies the number of bits to be extracted, is a negative value. The argument must be an integer in the range 0 - 63.

User Action: Be sure that you have specified the correct argument and that the argument is an integer in the range 0 - 63 inclusive.

Further Information: Refer to the description of the EXTB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 259-- {text}(arg2={text},arg3={text}). Starting bit must be < 64.

Condition Identifier: MLE\$EXTB_ARG2_RANGE Code: ML 259

Description: Note: This message is also associated with FORTRAN condition code FL619.

The second argument passed to EXTB, which specifies the first bit to be extracted, is greater than 63.
The argument must be an integer in the range 0 - 63.

User Action: Be sure that the argument is an integer in the range 0 -63 inclusive and that you have specified the correct argument.

Further Information: Refer to the description of the EXTB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 260-- {text}(arg2={text},arg3={text}). Starting bit + Length must be <= 64.

Condition Identifier: MLE\$EXTB_RANGE Code: ML 260

Description: Note: This message is also associated with FORTRAN condition code FL620.

The sum of the starting bit position (second argument to EXTB) and the number of bits to be extracted (third argument to EXTB) is greater than 64.

User Action: Be sure that the sum of the second and third arguments to EXTB does not exceed 64.

Further Information: Refer to the description of the EXTB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 261-- {text}(arg2={text},arg3={text}). Starting bit must be >= 0.

Condition Identifier: MLE\$INSB_ARG2_NEG Code: ML 261

Description: Note: This message is also associated with FORTRAN condition code FL621.

The second argument passed to INSB, which specifies the first bit to be inserted, is a negative value.
The argument must be an integer in the range 0 - 63.

User Action: Be sure that you have specified the correct argument and that the argument is an integer in the range 0 - 63 inclusive.

Further Information: Refer to the description of the INSB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 262-- {text}(arg2={text},arg3={text}). Length must be >= 0.

Condition Identifier: MLE\$INSB_ARG3_NEG Code: ML 262

Description: Note: This message is also associated with FORTRAN condition code FL622.

The third argument passed to INSB, which specifies the number of bits to be inserted, is a negative value. The argument must be an integer in the range 0 - 63.

User Action: Be sure that you have specified the correct argument and that the argument is an integer in the range 0 - 63 inclusive.

Further Information: Refer to the description of the INSB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 263-- {text}(arg2={text},arg3={text}). Starting bit must be < 64.

Condition Identifier: MLE\$INSB_ARG2_RANGE Code: ML 263

Description: This message is also associated with FORTRAN condition code FL 623.

The second argument passed to INSB, which specifies the first bit to be inserted, is greater than 63.
The argument must be an integer in the range 0 - 63.

User Action: Be sure that you have specified the correct argument and that the argument is an integer in the range 0 - 63 inclusive.

Further Information: Refer to the description of the INSB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 264-- {text}(arg2={text},arg3={text}). Starting bit + Length must be <= 64.

Condition Identifier: MLE\$INSB_RANGE Code: ML 264

Description: This message is also associated with FORTRAN condition code FL624.

The sum of the starting bit position (second argument to INSB) and the number of bits to be inserted (third argument to INSB) is greater than 64.

User Action: Be sure that the sum of the second and third arguments to INSB does not exceed 64.

Further Information: Refer to the description of the INSB routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 265-- {text}(0.0). Argument must be nonzero.

Condition Identifier: MLE\$COTAN_ARG_0 Code: ML 265

Description: This message is also associated with FORTRAN condition code FL629.

The argument passed to COTAN must be nonzero.

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the zero value was generated.

Further Information: Refer to the description of the COTAN routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 266-- {text}(arg1={text},arg2={text}). Arg1 must be zero if arg2 infinite.

Condition Identifier: MLE\$ATAN2_ARG2_RANGE Code: ML 266

Description: Note: This message is also associated with FORTRAN condition code FL630.

The first argument in the call to ATAN2 has an infinite value, and the second argument was nonzero. Although ATAN2 accepts an infinite value for the first argument if the second argument is zero, an infinite value is usually caused by an error somewhere in the program.

An infinite value is generated when a calculation produces a mathematically infinite result, such as an operation that involves very small or very large numbers, or division by zero.

User Action: Check all previous uses of both arguments to determine if an infinite value was generated. Check for improperly defined variables, variables that contain the incorrect data type, use of very small or very large numbers, bad input data, or a zero divisor.

Further Information: Refer to the description of the ATAN2 routine in the FORTRAN Language Definition Usage Manual.

--FATAL ML 267-- {text}({text},{text}). real part must be > -4097.*LOG(2).

Condition Identifier: MLE\$CEXP_REAL_RANGE2 Code: ML 267

Description: The absolute value of the real part of the argument passed to CEXP must be greater than -4097.*LOG(2). (-4097.*LOG(2) is approximately -2839.8240.)

User Action: Be sure that you have specified the correct argument. Then check all previous uses of the argument to determine how the incorrect value was generated.

Further Information: Refer to the description of the CEXP routine in the FORTRAN Language Definition Usage manual.

--FATAL MM 6-- Invalid PVA specified on the request.

Condition Identifier: MME\$INVALID_PVA Code: MM 6

Description: A PVA specified on the request is invalid.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 9-- Tried to read beyond EOI on read only segment - PVA = {text}, P = {text}.

Condition Identifier: MME\$READ_BEYOND_EOI Code: MM 9

Description: An attempt was made to read beyond the current end-of-information of the segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 10-- Uncorrected IO error - PVA = {text}, P = {text}.

Condition Identifier: MME\$IO_READ_ERROR Code: MM 10

Description: A media error occurred during the request.

User Action: Retry the operation. If this is not successful, recreate or reload the file.

--FATAL MM 11-- Tried to read/write beyond maximum segment length - PVA = {text}, P = {text}.

Condition Identifier: MME\$READ_WRITE_BEYOND_MSL Code: MM 11

Description: An attempt was made to either read or write beyond the current maximum segment length. The maximum segment length is the value specified by the system attribute, MAXIMUM_SEGMENT_LENGTH.

User Action: Determine the current segment offset, and the current value of the system attribute, MAXIMUM_SEGMENT_LENGTH. If necessary, it is possible to raise the system attribute to a higher value (assuming the hardware limit is not exceeded). It is only possible to lower the system attribute at the time of system core commands during deadstart.

--FATAL MM 14-- Not validated to access a segment from the present ring, PVA = {text}, P = {text}.

Condition Identifier: MME\$RING_VIOLATION Code: MM 14

Description: The validation ring number was insufficient for the type of segment access requested.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 15-- Invalid ring brackets in the SDTE.

Condition Identifier: MME\$INVALID_RING_BRACKETS Code: MM 15

Description: The ring values specified for ring one or ring two in the SEGMENT_ATTRIBUTE parameter are improper.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 16-- Attempt to set the binding attribute from above ring 3.

Condition Identifier: MME\$BINDING_ATTRIBUTE_INVALID Code: MM 16

Description: An attempt was made to set the binding segment attribute from above ring three.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 17-- Execute_global attribute can not be specified.

Condition Identifier: MME\$EXECUTE_GLOBAL_INVALID Code: MM 17

Description: An attempt was made to set the EXECUTE_GLOBAL attribute from above ring one.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 18-- Attempt to specify software attribute from beyond ring 3.

Condition Identifier: MME\$SOFTWARE_ATTRIBUTE_INVALID Code: MM 18

Description: An attempt was made to change the software attribute set from above ring three.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface. Several of the attributes are permissible to modify from above ring three.

--FATAL MM 20-- Attempt to close/delete segment that is not in callers write bracket or segment that is not a user segment.

Condition Identifier: MME\$INVALID_CLOSE_SEGMENT_REQ Code: MM 20

Description: An attempt was made to either close a segment other than a user segment, or the validation ring number specified on the request was insufficient.

User Action: Verify the segment origin of the segment, and check the validation ring number specified on the request.

--FATAL MM 21-- Caller not in read bracket of segment specified.

Condition Identifier: MME\$CALLER_NOT_IN_READ_BRACKET Code: MM 21

Description: The ring bracket of the caller is greater than the read bracket of the segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 22-- Caller not in write bracket of segment specified.

Condition Identifier: MME\$CALLER_NOT_IN_WRITE_BRACKET Code: MM 22

Description: The validation ring number of the caller is greater than the write bracket of the segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 23-- Attempt to set EXECUTE_LOCAL attribute from above ring 3.

Condition Identifier: MME\$EXECUTE_LOCAL_INVALID Code: MM 23

Description: Either an attempt was made to set the EXECUTE_LOCAL attribute with insufficient privilege, or an attempt was made to set the segment access control execute privilege to local privilege.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 24-- Attempt to change segment attribute that can not be modified.

Condition Identifier: MME\$SET_UNMODIFIABLE_ATTRIBUTE Code: MM 24

Description: An attempt was made to modify a segment attribute which can not be modified by this request.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 25-- Segment table is full.

Condition Identifier: MME\$SEGMENT_TABLE_IS_FULL Code: MM 25

Description: The segment table of the task is full. A maximum of 4096 segments can be in task's segment table.

User Action: Delete extraneous segments from the task, and repeat the request.

--FATAL MM 26-- Segment number specified is already in use.

Condition Identifier: MME\$SEGMENT_NUMBER_IS_IN_USE Code: MM 26

Description: An attempt was made to use a segment number which is already in use.

User Action: Reissue the request specifying a different segment number. If this is not possible, the user may attempt to delete the segment occupying the segment number position which is desired. Another option may be to change/move the segment in the desired segment number, and the change/move the original segment to the recently vacated segment number.

--FATAL MM 27-- Segment number specified is not in use.

Condition Identifier: MME\$SEGMENT_NUMBER_NOT_IN_USE Code: MM 27

Description: The segment number specified on the request is not currently valid.

User Action: Reissue the request, specifying a valid segment number. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 28-- Segment number is beyond the range of the segment table.

Condition Identifier: MME\$SEGMENT_NUMBER_TOO_BIG Code: MM 28

Description: The segment number specified on the request is larger than the maximum segment number allowed for a task. The maximum number of segments in a task is 4096.

User Action: Reissue the request, specifying a valid segment number. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 29-- Keyword specified for the segment attributes is invalid.

Condition Identifier: MME\$UNSUPPORTED_KEYWORD Code: MM 29

Description: One or more of the attributes specified on the request is not a valid attribute.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 30-- Attempt to create inherited SDT and SDT or SDTX already exists.

Condition Identifier: MME\$SDT_OR_SDTX_EXIST Code: MM 30

Description: During task initiation, a segment table or segment table extended was found for this task. At this point in task initiation, neither of these tables should exist.

User Action: Terminate the task which was being created, and reattempt the task.

--FATAL MM 32-- ASID specified as attribute and segment manager not called from ring 1.

Condition Identifier: MME\$ASID_SPECIFIED Code: MM 32

Description: An attempt was made to specify an ASID with insufficient privilege. A validation ring number of one is required, if an ASID is specified.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 33-- The specified ASID is not one of the reserved ASID's.

Condition Identifier: MME\$INVALID_ASID_SPECIFIED Code: MM 33

Description: The ASID specified on the request is not one of the ASIDs which is reserved.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 35-- Attempt to lock a page that is already locked.

Condition Identifier: MME\$PAGE_ALREADY_LOCKED Code: MM 35

Description: A request was issued to lock a page which is already locked by this task.

User Action: Correct the program logic so that it cannot request to lock a page which it has already locked.
Refer to the interface documentation for further information on locking pages.

--FATAL MM 36-- Attempt to do IO on a segment with no backing file.

Condition Identifier: MME\$SEGMENT_NOT_ASSIGNED_DEVICE Code: MM 36

Description: A request was made to perform IO on a segment which does not have backing storage assigned.
The segment must be assigned to a disk before IO can be performed on the segment.

User Action: Insure that the segment has been assigned to a disk before attempting any IO operations.
Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 40-- The taskid specified on the call is invalid.

Condition Identifier: MME\$INVALID_TASK_ID Code: MM 40

Description: During task creation, the task id of the new task was found to be NIL.

User Action: Terminate the task, and then retry the task.

--FATAL MM 42-- Invalid request code.

Condition Identifier: MME\$INVALID_REQUEST Code: MM 42

Description: There was an attempt to perform a operation which is not valid for this request.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 43-- Io error on trying to write a page to disk.

Condition Identifier: MME\$IO_WRITE_ERROR Code: MM 43

Description: While attempting to write the page or pages to disk, IO errors were detected.

User Action: Reissue the request.

--FATAL MM 44-- Attempt to age out a page in a non pageable segment.

Condition Identifier: MME\$SEGMENT_NOT_PAGEABLE Code: MM 44

Description: An attempt was made to age out a page or pages from a segment which is not pageable.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 45-- Attempt to set segment origin from above ring 1.

Condition Identifier: MME\$SEGMENT_ORIGIN_INVALID Code: MM 45

Description: An attempt was made to modify the origin of a segment from above ring one.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 46-- Attempt to modify segment origin.

Condition Identifier: MME\$SEGMENT_ORIGIN_CHANGE Code: MM 46

Description: An attempt was made to modify the segment origin. This is not a valid operation with this request.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 47-- Lock/unlock request and length + offset > maximum segment length.

Condition Identifier: MME\$LOCK_UNLOCK_INVALID_LENGTH Code: MM 47

Description: An attempt was made to lock or unlock pages beyond maximum segment size. The sum of the requested length and the current offset of the segment exceeds the maximum segment size.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 51-- Set segment length on segment without write access.

Condition Identifier: MME\$NO_WRITE_ACCESS Code: MM 51

Description: An attempt was made to modify the segment length for a segment which the user does not have write access.

User Action: Refer to the request documentation to determine the request restrictions.

--FATAL MM 53-- Write beyond segment eoi with no append permission - PVA = {text}, P = {text}.

Condition Identifier: MME\$WRITE_BEYOND_EOI_NO_APPEND Code: MM 53

Description: An attempt was made to write beyond the segment's current end-of-information by a user which does not have append permission for the segment.

User Action: Correct the request to not attempt to write beyond the segment's end-of-information, or modify the segment access to include extend permission.

--FATAL MM 55-- Segment already locked by current task

Condition Identifier: MME\$SEGMENT_LOCKED_BY_TASK Code: MM 55

Description: A request was made to lock a segment by a task which already has the segment locked.

User Action: Correct the program logic so that it does not request locks which it already has.

--FATAL MM 56-- Segment locked by another task

Condition Identifier: MME\$SEGMENT_LOCKED_ANOTHER_TASK Code: MM 56

Description: An attempt was made to lock a segment which is currently locked by another task.

User Action: Reissue the request. Also, refer to the request documentation's discussion on the WAIT_OPTION parameter. If the WAIT_OPTION parameter is set to TRUE, the request will not return until it has successfully obtained the lock.

--FATAL MM 57-- Segment not locked by current task

Condition Identifier: MME\$SEGMENT_NOT_LOCKED Code: MM 57

Description: A request was issued to unlock a segment which is not currently locked by this task.

User Action: Correct the program logic so that it does not issue requests to unlock segments which it does not currently have locked.

--FATAL MM 65-- Attempt to check status of io with pointers for which no io has been requested.

Condition Identifier: MME\$INVALID_IO_STATUS_PTRS Code: MM 65

Description: The pointers passed into this request have not had IO requested. The pointers passed into this request must be the same pointers used in the IO request.

User Action: Review the values specified for the request parameters. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 68-- Length on read/write request exceeds 65536 bytes.

Condition Identifier: MME\$REQUEST_LENGTH_TOO_LONG Code: MM 68

Description: The value specified for the LENGTH parameter exceeds the maximum length allowed for this request. The maximum length allowed for this request is 65536 bytes.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 69-- Offset plus length created an invalid pva.

Condition Identifier: MME\$INVALID_PVA_FORMED Code: MM 69

Description: The value specified in the LENGTH parameter plus the current offset of the PVA specified exceed the maximum segment length.

User Action: Review the values specified for the request parameters. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 70-- The file backing the segment being referenced has not been recovered.

Condition Identifier: MME\$REF_TO_UNRECOVERED_FILE Code: MM 70

Description: The backing file associated with this segment has not been recovered.

User Action: Please submit a PSR. Any information that you can supply about the file would be beneficial. Also, include the dump from the most recent system failure if possible.

--FATAL MM 71-- The length of the shadow file must be a multiple of 16384.

Condition Identifier: MME\$LENGTH_NOT_0_MOD_16384 Code: MM 71

Description: The length of the shadow segment is not a multiple of 16384.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 72-- The starting address of the shadow file must be a multiple of 16384.

Condition Identifier: MME\$ADDRESS_NOT_0_MOD_16384 Code: MM 72

Description: The offset of the shadow segment pointer is not a multiple of 16384.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 73-- The specified shadow file is not valid.

Condition Identifier: MME\$INVALID_SHADOW_SEGMENT Code: MM 73

Description: The shadow segment specified in the SEGMENT_ATTRIBUTES parameter of the request is either not a valid segment, or the segment specified does not have a backing file assigned.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 74-- The segment specified is not appropriate for shadowing.

Condition Identifier: MME\$INIT_SHADOW_IMPROPER_SEG Code: MM 74

Description: The shadow segment specified on the request either had improper ring attributes, or was a writable segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 76-- The use of wired or fixed segments prohibited with this request.

Condition Identifier: MME\$WIRED_OR_FIXED_SEGS_ILLEGAL Code: MM 76

Description: The PVA specified was that of either a wired or fixed segment. This request does not support these types of segments.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 78-- Memory is not currently available for assign_pages request.

Condition Identifier: MME\$MEMORY_NOT_AVAIL_FOR_ASSIGN Code: MM 78

Description: The request can not be honored at this time because there is not enough memory available in the system.

User Action: Reissue the request at some time in the future. It is possible to specify TRUE on the WAIT_OPTION parameter of the request, and the request will not complete until it successfully obtains the requested memory. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 80-- The length requested would cause the working set to get too large.

Condition Identifier: MME\$ASSIGN_LENGTH_TOO_LONG Code: MM 80

Description: The value specified for the LENGTH parameter was too large. It would either cause the working set of the job to exceed the maximum working set for this job, or the file limit would be exceeded.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 81-- The requested length must be positive.

Condition Identifier: MME\$LENGTH_MUST_BE_POSITIVE Code: MM 81

Description: The value of the LENGTH parameter specified was either less than or equal to zero.

User Action: Review the value specified for the LENGTH parameter. Reissue the request specifying a valid length. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 83-- Job is non swappable -- cannot wait to assign memory.

Condition Identifier: MME\$CANNOT_WAIT_FOR_MEMORY Code: MM 83

Description: Currently, there is not enough memory to honor the request. Even though the WAIT_OPTION parameter was set to TRUE, the job is non-swappable, and we can not wait for memory in this situation.

User Action: Retry the request at some point in the future. It is impossible to determine when the memory will become available.

--FATAL MM 84-- An illegal attempt was made to change the segment origin.

Condition Identifier: MME\$ILLEGAL_SEGMENT_ORIGIN_CHG Code: MM 84

Description: An attempt was made to change the segment origin of a segment whose original segment origin was not mmc\$so_user.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 85-- An illegal taskid was found either opening or closing a shared stack segment.

Condition Identifier: MME\$INVALID_SHARED_TASKID Code: MM 85

Description: One or more of the taskids specified on the request are invalid.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 86-- A reference has been made to a segment on a volume that is not available.

Condition Identifier: MME\$VOLUME_UNAVAILABLE Code: MM 86

Description: An attempt was made to access a segment which is located on volume which is currently unavailable.

User Action: Retry the request when the volume becomes available.

--FATAL MM 88-- The requested length of the wired segment exceeds 65536 bytes.

Condition Identifier: MME\$WIRED_SEG_LENGTH_TOO_LARGE Code: MM 88

Description: The segment length specified in the SEGMENT_ATTRIBUTE parameter of this request exceeds the 65536 limit.

User Action: Review the segment length specified, and reissue the request with a valid segment length specified. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 90-- The requested length on mmp\$move_pages must be a page size multiple.

Condition Identifier: MME\$LENGTH_NOT_PAGE_SIZE_MULT Code: MM 90

Description: The value specified for the LENGTH parameter of this request is not a multiple of the page size currently being used by the system.

User Action: Review the value specified for the LENGTH parameter on the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 91-- The specified pvas on mmp\$move_pages must be on a page boundary.

Condition Identifier: MME\$PVA_NOT_ON_PAGE_BOUNDARY Code: MM 91

Description: Either the offset of the source page PVA or the calculated offset of the destination page was not on a page boundary.

User Action: Review the parameter values specified on the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 93-- Source page was modified on mmp\$move_pages_request.

Condition Identifier: MME\$MODIFIED_SOURCE_PAGE_REJECT Code: MM 93

Description: A source page was detected as modified while processing this request. Because the REJECT_MOVE_IF_SOURCE_MODIFIED parameter was TRUE, the request failed.

User Action: Review the parameters specified on the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 95-- Length is greater than maximum allowed or less than minimum allowed.

Condition Identifier: MME\$INVALID_LENGTH_REQUESTED Code: MM 95

Description: The value specified for the LENGTH parameter was invalid. It was either less than or equal to zero, or greater than the request limit.

User Action: Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 97-- Interface does not support segments of this kind--must be mmc\$sk_file.

Condition Identifier: MME\$UNSUPPORTED_SEGMENT_KIND Code: MM 97

Description: The segment specified in the segment pointer parameter does not have a backing file assigned. This is not a valid segment kind for this request.

User Action: Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 100-- Segment must be either wired or job_fixed to assign contiguous memory.

Condition Identifier: MME\$CONTIG_MEM_SEG_VIOLATION Code: MM 100

Description: An attempt was made to assign contiguous memory to a segment which was not a wired or job-fixed segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 101-- Unable to allocate the requested amount of contiguous memory.

Condition Identifier: MME\$UNABLE_TO_ASSIGN_CONTIG_MEM Code: MM 101

Description: The request can not be completed because there are not enough contiguous pages available at the present time.

User Action: Continue to retry the request until it is successful. It is impossible to tell when enough memory will be available to honor the request.

--FATAL MM 102-- Pages within the range (PVA-->PVA+length) are already assigned.

Condition Identifier: MME\$PAGES_ALREADY_ASSIGNED Code: MM 102

Description: One or more of the pages within the range (PVA --> PVA + length) are currently assigned.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 103-- Updating the passive segment requires write permission.

Condition Identifier: MME\$UPDATE_REQ_WRITE_PERMISSION Code: MM 103

Description: To update the passive segment during shadow file termination, write permission is required for the segment.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 105-- Transient segments can not be shadowed.

Condition Identifier: MME\$CANT_SHADOW_TRANSIENT_SEGS Code: MM 105

Description: An attempt was made to shadow a segment which had a segment kind which is illegal for this request.

User Action: Review the values of the parameters specified in the request. Refer to the request documentation to determine the restrictions which are imposed on the individual parameters of this interface.

--FATAL MM 106-- The segment is located on a terminated file_server and therefore it cannot be accessed - PVA = {text}, P = {text}.

Condition Identifier: MME\$FILE_SERVER_TERMINATED Code: MM 106

Description: An attempt was made to access a segment which is located on a terminated file server.

User Action: Attempt the request again sometime later.

--ERROR MV 10-- You are not authorized to use the MAIL/VE system.

Condition Identifier: MVE\$USER_NOT_REGISTERED Code: MV 10

Description: Users of the MAIL/VE system must be registered by the MAIL/VE system administrator.

User Action: Contact your MAIL/VE system administrator.

--INFORMATIVE MV 11-- You have no private mailbox.

Condition Identifier: MVE\$NO_PRIVATE_MAILBOX Code: MV 11

Description: The mail system did not create a private mailbox for you when you were validated on the system because of a conflict between your user name and existing mailbox name.

User Action: Use the CREATE_MAILBOX command to create a private mailbox.

--ERROR MV 12-- User name not found in User Directory File.

Condition Identifier: MVE\$UNKNOWN_USER Code: MV 12

Description: The MAIL/VE system administrator attempted to reference or delete a user who is not registered.

User Action: None.

--ERROR MV 13-- User name already registered in User Directory File.

Condition Identifier: MVE\$USER_ALREADY_REGISTERED Code: MV 13

Description: The MAIL/VE system administrator has attempted to register a user that was already registered.

User Action: None.

--ERROR MV 14-- User requested interrupt, control-p.

Condition Identifier: MVE\$USER_TYPED_CONTROL_P Code: MV 14

Description: The user entered control-p while running the mail system in screen mode. The current command is terminated and the mail system terminates.

User Action: Enter ACTIVATE_SCREEN to return to screen mode.

--ERROR MV 15-- User requested abort, control-t.

Condition Identifier: MVE\$USER_TYPED_CONTROL_T Code: MV 15

Description: The user entered control-t while running the mail system in screen mode. The current command is terminated and the mail system terminates.

User Action: Enter ACTIVATE_SCREEN to return to screen mode.

--ERROR MV 16-- MAIL/VE system is not available, maintenance in progress. Try again later.

Condition Identifier: MVE\$MAIL_NOT_AVAILABLE Code: MV 16

Description: The MAIL/VE system administrator has control of the system for maintenance.

User Action: Wait until the MAIL/VE administrator has completed system maintenance and try again.

--ERROR MV 17-- MAIL/VE is currently in use. All users must be logged out of MAIL/VE.

Condition Identifier: MVE\$MAIL_SYSTEM_ACTIVE Code: MV 17

Description: The MAIL/VE system administrator attempted to execute a command that requires exclusive use of MAIL/VE files while other users are logged into MAIL/VE.

User Action: Wait until all users have logged out of MAIL/VE.

--ERROR MV 18-- MAIL/VE utility is already active.

Condition Identifier: MVE\$SECOND_CALL_TO_MAIL Code: MV 18

Description: The MAIL utility may not be called again from within an active MAIL session.

User Action: None.

--ERROR MV 19-- MAIM must be run from the MAIL/VE master catalog.

Condition Identifier: MVE\$MAIM_RUNS_IN_MAIL_CATALOG Code: MV 19

Description: The MAIL/VE maintenance commands are only valid from the MAIL/VE master catalog.

User Action: None.

--ERROR MV 20-- MAIM must be run by the MAIL/VE System Administrator.

Condition Identifier: MVE\$NOT_SYSTEM_ADMINISTRATOR Code: MV 20

Description: The MAIL/VE maintenance commands may only be run by the system administrator.

User Action: Not applicable.

--ERROR MV 21-- The terminal model specified is not recognized by MAIL/VE. Use SETTA TRM= to select a standard model, or a combination of SETTA TRM= and SETPA AL= to add a library containing the appropriate terminal definition module for your terminal to your library list.

Condition Identifier: MVE\$NOT_RECOGNIZED_MODEL Code: MV 21

Description: MAIL/VE does not recognize the terminal model specified.

User Action: Use the SET_TERMINAL_ATTRIBUTES TERMINAL_MODEL parameter to select a standard model, or a combination of this command and the SET_PROGRAM_ATTRIBUTE ADD_LIBRARIES parameter to add a library containing the appropriate terminal definition module for your terminal to your library list.

--FATAL MV 40-- Family Directory File is empty.

Condition Identifier: MVE\$NO_FAMILIES Code: MV 40

Description: The MAIL/VE system administrator has attempted a command requiring a NOS/VE family name but no families have been defined.

User Action: The MAIL/VE system administrator must define NOS/VE family names and then repeat this command.

--ERROR MV 43-- Family name not found in Family Directory File.

Condition Identifier: MVE\$FAMILY_NAME_NOT_FOUND Code: MV 43

Description: The NOS/VE family name has not been defined by the MAIL/VE system administrator.

User Action: The MAIL/VE system administrator must define the NOS/VE family name before referencing the family in other commands.

--ERROR MV 45-- Attempt to add a family that already exists.

Condition Identifier: MVE\$FAMILY_NAME_IN_USE Code: MV 45

Description: The NOS/VE family name is already known to the mail system.

User Action: None.

--ERROR MV 62-- A mailbox or distribution list by that name already exists.

Condition Identifier: MVE\$ADDRESSEE_KEY_IN_USE Code: MV 62

Description: Names used to create mailboxes and distribution lists must be unique within a NOS/VE family.

User Action: Select a different mailbox or distribution list name and reissue the command.

--ERROR MV 71-- Attempt to add owned mailbox name that already exists.

Condition Identifier: MVE\$DUPLICATE_OWNED_MAILBOX Code: MV 71

Description: Mailbox names must be unique within a NOS/VE family. The user currently owns a mailbox with this name.

User Action: Not applicable.

--ERROR MV 72-- Attempt to add PM mailbox which is already permitted.

Condition Identifier: MVE\$DUPLICATE_PM_MAILBOX Code: MV 72

Description: A group mailbox with this name is already owned by the user. Mailbox names must be unique within a NOS/VE family.

User Action: None.

--ERROR MV 73-- A mailbox appears more than once in the Permitted Mailbox list.

Condition Identifier: MVE\$DUPLICATE_PM_IN_LIST Code: MV 73

Description: The PERMITTED_MAILBOX parameter contains a repeated mailbox name.

User Action: Remove duplicate mailbox names from the PERMITTED_MAILBOX parameter and reissue the command.

--ERROR MV 74-- User has reached limit on number of owned mailboxes.

Condition Identifier: MVE\$TOO_MANY_OWNED_MAILBOXES Code: MV 74

Description: A user may own a maximum of 256 mailboxes.

User Action: None.

--ERROR MV 75-- Attempt to reference (delete) non-existent owned mailbox.

Condition Identifier: MVE\$OWNED_MAILBOX_NOT_FOUND Code: MV 75

Description: The user attempted to reference or delete a mailbox that he owned but has subsequently deleted.

User Action: None.

--ERROR MV 76-- Attempt to reference (delete) non-existent PM mailbox.

Condition Identifier: MVE\$PM_MAILBOX_NOT_FOUND Code: MV 76

Description: The user attempted to reference or delete a mailbox that he had been permitted to use but which has subsequently been deleted by its owner.

User Action: None.

--ERROR MV 77-- Attempt to add new permitted mailbox to a group mailbox when limit reached.

Condition Identifier: MVE\$TOO_MANY_PM_MAILBOXES Code: MV 77

Description: A group mailbox may contain a maximum of 256 private mailboxes.

User Action: None.

--ERROR MV 82-- Attempt to add a mailbox which already exists.

Condition Identifier: MVE\$MAILBOX_KEY_IN_USE Code: MV 82

Description: Names used to create mailboxes and distribution lists must be unique within a NOS/VE family.

User Action: Select a different mailbox name and reissue the command.

--ERROR MV 83-- This operation is only allowed for group mailboxes.

Condition Identifier: MVE\$NOT_GROUP_MAILBOX Code: MV 83

Description: The current command is only valid for group mailboxes.

User Action: Not applicable.

--ERROR MV 84-- This operation is only allowed for private mailboxes.

Condition Identifier: MVE\$NOT_PRIVATE_MAILBOX Code: MV 84

Description: The current command is only valid for private mailboxes.

User Action: None.

--ERROR MV 85-- Attempt to add a second private mailbox not allowed.

Condition Identifier: MVE\$ONLY_ONE_PRIVATE_MAILBOX Code: MV 85

Description: Each user is limited to one private mailbox.

User Action: Not appli

--ERROR MV 86-- Attempt to add list of permitted mailboxes to a non-group mailbox.

Condition Identifier: MVE\$ONLY_GROUP_CAN_HAVE_LIST Code: MV 86

Description: Only group mailboxes may have a permitted mailbox list.

User Action: None.

--ERROR MV 87-- Attempt to delete a private mailbox.

Condition Identifier: MVE\$CANNOT_DELETE_PRIVATE_MB Code: MV 87

Description: Only the MAIL/VE system administrator can delete a private mailbox.

User Action: Contact the MAIL/VE system administrator to delete private mailboxes.

--ERROR MV 88-- You may do that function only if you own the mailbox.

Condition Identifier: MVE\$MAILBOX_NOT_OP_OWNED Code: MV 88

Description: A mailbox may only be deleted or changed by the owner or creator of the mailbox.

User Action: None.

--INFORMATIVE MV 90-- Selection Criteria produced no mailboxes to display.

Condition Identifier: MVE\$BUILD_DISM_NONE_FOUND Code: MV 90

Description: No mailbox satisfies the search criteria specified on the DISPLAY_MAILBOX command.

User Action: Not applicable.

--INFORMATIVE MV 110-- Selection Criteria produced no letters to display.

Condition Identifier: MVE\$BUILD_DISL_NONE_FOUND Code: MV 110

Description: No letter satisfies the search criteria specified on the DISPLAY_LETTER command.

User Action: None.

--ERROR MV 116-- Cannot SEND/GET letters from files of this file organization type.

Condition Identifier: MVE\$INVALID_LETTER_FILE_ORG Code: MV 116

Description: Only sequential and index sequential file organization types may be used to send or get letters.

User Action: Use sequential or index sequential file organization for sending and receiving letters.

--ERROR MV 119-- This letter has already been deleted.

Condition Identifier: MVE\$LETTER_DELETED Code: MV 119

Description: The user has deleted this letter number since he last did a DISPLAY_LETTER.

User Action: Use DISPLAY_LETTER to display current letter numbers.

--ERROR MV 153-- Wildcard specification has invalid character sequence.

Condition Identifier: MVE\$INVALID_WILDCARD_NAME Code: MV 153

Description: The wildcard specification contains characters or a string format that is not a valid mailbox or distribution list name.

User Action: None.

--ERROR MV 155-- Letter number out of range.

Condition Identifier: MVE\$NUMBER_OUT_OF_RANGE Code: MV 155

Description: The letter number specified does not correspond to any letter number assigned by the DISPLAY_LETTER command.

User Action: Use the DISPLAY_LETTER command to check valid letter numbers.

--ERROR MV 156-- User does not have permission to access this mailbox.

Condition Identifier: MVE\$NOT_PERMITTED_MAILBOX Code: MV 156

Description: You may only access your own private mailbox, public mailboxes, and group mailboxes to which you are permitted access.

User Action: Not applicable.

--ERROR MV 158-- This function requires a previous DISPLAY_LETTER command.

Condition Identifier: MVE\$NO_ACTIVE_DISL Code: MV 158

Description: The DISPLAY_LETTER command must precede other commands, such as GET_LETTER, that use the numbered letter list.

User Action: Execute the DISPLAY_LETTER command and then reissue this command.

--ERROR MV 160-- Entered value for SUBJECT parameter is in error.

Condition Identifier: MVE\$ILLEGAL_SUBJECT Code: MV 160

Description: A letter's subject must be a string of up to 72 characters.

User Action: None.

--ERROR MV 171-- Attempt to create a new Distribution List which already exists.

Condition Identifier: MVE\$DL_KEY_IN_USE Code: MV 171

Description: Distribution list names must be unique within a NOS/VE family. The name is currently in use as a mailbox or distribution list name.

User Action: None.

--ERROR MV 172-- A Distribution List can have no more than 65535 subordinate mailboxes.

Condition Identifier: MVE\$TOO_MANY_1ST_LEVEL_SUBORDS Code: MV 172

Description: The distribution list referenced directly and indirectly more than the 65535 maximum number of mailboxes.

User Action: None.

--ERROR MV 173-- A distribution list references a mailbox which does not exist.

Condition Identifier: MVE\$IMPROPER_MAILBOX_IN_DL Code: MV 173

Description: A directly or indirectly referenced distribution list contains a mailbox name that has been deleted from the system.

User Action: Check the distribution lists for invalid mailbox names.

--ERROR MV 174-- A distribution list references another which is not known.

Condition Identifier: MVE\$IMPROPER_DL_IN_DL Code: MV 174

Description: A directly or indirectly referenced distribution list contains a distribution list name that has been deleted from the system.

User Action: Check the distribution lists for invalid distribution list names.

--INFORMATIVE MV 177-- Selection Criteria produced no distribution lists to display.

Condition Identifier: MVE\$BUILD_DISDL_NONE_FOUND Code: MV 177

Description: No distribution list satisfies the search criteria specified on the DISPLAY_DISTRIBUTION_LIST command.

User Action: None.

--ERROR MV 178-- Distribution List not found.

Condition Identifier: MVE\$DISTRIBUTION_LIST_NOT_FOUND Code: MV 178

Description: The named distribution list does not exist.

User Action: None.

--ERROR MV 179-- LIST_NAME is not a distribution list.

Condition Identifier: MVE\$NOT_A_DISTRIBUTION_LIST Code: MV 179

Description: The name specified for the LIST_NAME parameter is not a distribution list.

User Action: None.

--ERROR MV 191-- Only the owner can change or delete a distribution list.

Condition Identifier: MVE\$D_LIST_NOT_OP_OWNED Code: MV 191

Description: The distribution list was not created and therefore not owned by the current user.

User Action: None.

--ERROR MV 192-- Unknown distribution list member.

Condition Identifier: MVE\$UNKNOWN_DL_MEMBER Code: MV 192

Description: A mailbox name in the distribution list is not known to the MAIL/VE system.

User Action: Not applicable.

--ERROR MV 193-- Public distribution list cannot contain a private list.

Condition Identifier: MVE\$PRIVATE_DL_NOT_ALLOWED Code: MV 193

Description: A public distribution list may only contain mailboxes and other public distribution lists.

User Action: None.

--ERROR MV 194-- Only the owner can use a private distribution list.

Condition Identifier: MVE\$CANNOT_USE_PRIVATE_LIST Code: MV 194

Description: You may only use your own private mailbox, public mailboxes, and group mailboxes to which you are permitted access.

User Action: None.

--ERROR NA 6-- Number of SAP activities is at zero.

Condition Identifier: NAE\$ACTIVITY_COUNT_ZERO Code: NA 6

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 11-- Allocation failed for {text}.

Condition Identifier: NAE\$ALLOCATION_FAILED Code: NA 11

Description: An allocation failed because all available system resources are temporarily busy.

User Action: Retry later.

--ERROR NA 26-- Channelnet {text} not configured.

Condition Identifier: NAE\$CHANNELNET_NOT_CONFIGURED Code: NA 26

Description: Internal error.

User Action: Site analyst must add the network to the system configuration.

--ERROR NA 31-- The connection was terminated by the peer process.

Condition Identifier: NAE\$CONN_TERMINATED_BY_PEER Code: NA 31

Description: The connection was terminated by the peer process.

User Action: No action required.

--ERROR NA 36-- The connection was terminated due to a failure on the network.

Condition Identifier: NAE\$CONNECTION_FAILED Code: NA 36

Description: The connection was terminated due to a failure on the network.

User Action: No action required.

--ERROR NA 41-- The connection is full of outbound messages and cannot process a {text}.

Condition Identifier: NAE\$CONNECTION_FULL Code: NA 41

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 46-- {text} layer connection {text} not open.

Condition Identifier: NAE\$CONNECTION_NOT_OPEN Code: NA 46

Description: The connection requested is not open.

User Action: Correct user program.

--ERROR NA 51-- {text} layer connection {text} not proposed for ACCEPT/REJECT.

Condition Identifier: NAE\$CONNECTION_NOT_PROPOSED Code: NA 51

Description: The connection specified for ACCEPT/REJECT is not proposed.

User Action: Correct user program.

--ERROR NA 61-- The data area is too small. {text}

Condition Identifier: NAE\$DATA_AREA_TOO_SMALL Code: NA 61

Description: The user's data_area does not meet the requirements of the request.

User Action: Increase the size of the data_area.

--ERROR NA 101-- The aggregate event kind is not 'data' or 'expedited_data' on send_aggregate_data.

Condition Identifier: NAE\$IMPROPER_AGGREGATE_KIND Code: NA 101

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 121-- An incorrect activity is specified in the wait list.

Condition Identifier: NAE\$INCORRECT_ACTIVITY Code: NA 121

Description: An incorrect activity is specified in the wait list.

User Action: Remove incorrect wait list element.

--ERROR NA 126-- Buffer manager: Insufficient data to extract.

Condition Identifier: NAE\$INSUFFICIENT_DATA Code: NA 126

Description: Insufficient data to extract.

User Action: No action required.

--ERROR NA 131-- The requestor has insufficient privilege for the request.

Condition Identifier: NAE\$INSUFFICIENT_PRIVILEGE Code: NA 131

Description: The user is not authorized to make this request.

User Action: Correct the user's program to call authorized program interfaces.

--ERROR NA 141-- {text} invalid job.

Condition Identifier: NAE\$INVALID_JOB Code: NA 141

Description: The request can only be made by the job that opened the Service Access Point.

User Action: Correct user's program.

--ERROR NA 158-- {text} request cannot be accepted at this time. {text}

Condition Identifier: NAE\$INVALID_REQUEST Code: NA 158

Description: The request cannot be accepted at this time.

User Action: No action required.

--ERROR NA 161-- The reserved SAP specified is improper.

Condition Identifier: NAE\$INVALID_RESERVED_SAP Code: NA 161

Description: The SAP specified is not a reserved SAP.

User Action: Retry, specifying a SAP that is a reserved SAP.

--ERROR NA 171-- {text}.

Condition Identifier: NAE\$INVALID_TASK Code: NA 171

Description: The request can only be made by the task that opened the Service Access Point.

User Action: Correct user's program.

--ERROR NA 191-- {text} terminated due to job recovery.

Condition Identifier: NAE\$JOB_RECOVERY Code: NA 191

Description: The request has terminated due to job recovery.

User Action: No action required.

--ERROR NA 201-- Maximum active connections specified exceeds maximum value.

Condition Identifier: NAE\$MAX_ACTIVE_CONN_EXCEEDED Code: NA 201

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 211-- Number of active connections already at maximum.

Condition Identifier: NAE\$MAX_ACTIVE_CONNECTIONS Code: NA 211

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 221-- Maximum active connections specified is zero.

Condition Identifier: NAE\$MAX_ACTIVE_CONNECTIONS_0 Code: NA 221

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 231-- Maximum active count specified exceeds maximum value.

Condition Identifier: NAE\$MAX_ACTIVITY_COUNT_EXCEEDED Code: NA 231

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 241-- The number of outstanding expedited requests is at the limit for the connection.

Condition Identifier: NAE\$MAX_EXPEDITED_EXCEEDED Code: NA 241

Description: The number of outstanding expedited requests is at the limit for the connection.

User Action: Retry later.

--ERROR NA 251-- Number of SAP activities already at maximum.

Condition Identifier: NAE\$MAXIMUM_ACTIVITY_COUNT Code: NA 251

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 261-- Maximum number of {text} layer SAPs already open.

Condition Identifier: NAE\$MAXIMUM_SAPS_OPEN Code: NA 261

Description: The maximum number of Service Access Points is already open.

User Action: No action required.

--ERROR NA 301-- No datagram available for {text} layer SAP {text}.

Condition Identifier: NAE\$NO_DATAGRAM_AVAILABLE Code: NA 301

Description: No datagram available.

User Action: Retry later.

--ERROR NA 311-- {text} no event found.

Condition Identifier: NAE\$NO_EVENT Code: NA 311

Description: No event was found.

User Action: Retry later.

--ERROR NA 315-- The destination system is not reachable at this time.

Condition Identifier: NAE\$DESTINATION_NOT_REACHABLE Code: NA 315

Description: The Channel Connection connect request could not be sent to the requested device because it was unavailable.

User Action: Retry later when the device is available.

--ERROR NA 351-- {text} layer protocol error.

Condition Identifier: NAE\$PROTOCOL_ERROR Code: NA 351

Description: Internal error.

User Action: Notify your site analyst.

--ERROR NA 451-- The connection cannot be established due to unavailable system resources.

Condition Identifier: NAE\$RESOURCES_UNAVAILABLE Code: NA 451

Description: The requested connection cannot be established due to the unavailability of the network buffers or allocatable memory space.

User Action: Retry at a later time.

--ERROR NA 501-- Reserved SAP {text} is already in use.

Condition Identifier: NAE\$SAP_ALREADY_OPEN Code: NA 501

Description: The specified Service Access Point is already open.

User Action: No action required.

--ERROR NA 511-- {text} layer SAP {text} not open.

Condition Identifier: NAE\$SAP_NOT_OPEN Code: NA 511

Description: The specified Service Access Point is not open.

User Action: Open a SAP and retry request.

--ERROR NA 521-- The connection has reached the supervisory traffic limit and cannot process a {text} request.

Condition Identifier: NAE\$SUPERVISORY_TRAFFIC_LIMIT Code: NA 521

Description: The local application is sending interrupt requests and/or synchronize requests faster than the peer application can process the requests.

User Action: Use interrupt requests and synchronize requests judiciously.

--ERROR NA 601-- System interrupt.

Condition Identifier: NAE\$SYSTEM_INTERRUPT Code: NA 601

Description: A system interrupt caused the request to terminate.

User Action: No action required.

--ERROR NA 701-- Unable to open {text} sap.

Condition Identifier: NAE\$UNABLE_TO_OPEN_SAP Code: NA 701

Description: Unable to open Service Access Point. All system resources are busy.

User Action: Retry later.

--ERROR NA 1005-- Break condition active.

Condition Identifier: NAE\$BREAK_CONDITION_ACTIVE Code: NA 1005

Description: A break condition is active.

User Action: Retry later.

--ERROR NA 1010-- Connection active.

Condition Identifier: NAE\$CONNECTION_ACTIVE Code: NA 1010

Description: The requested file operation cannot be completed while the connection is active.

User Action: Retry later.

--ERROR NA 1030-- Interactive cond interrupt.

Condition Identifier: NAE\$INTERACTIVE_COND_INTERRUPT Code: NA 1030

Description: The request cannot be completed while an interactive condition interrupt is being processed.

User Action: Retry request.

--ERROR NA 1050-- Nominal connection.

Condition Identifier: NAE\$NOMINAL_CONNECTION Code: NA 1050

Description: The requested operation cannot be completed on a nominal connection.

User Action: Correct user programming error.

--ERROR NA 1055-- Not nominal connection.

Condition Identifier: NAE\$NOT_NOMINAL_CONNECTION Code: NA 1055

Description: The request can only be made on a nominal connection.

User Action: Correct user programming error.

--ERROR NA 1060-- Not simulated connection broken.

Condition Identifier: NAE\$NOT_SIMULATED_CONN_BROKEN Code: NA 1060

Description: The request can only be made while simulating a broken connection.

User Action: No action required.

--ERROR NA 1080-- Simulated connection broken.

Condition Identifier: NAE\$SIMULATED_CONNECT_BROKEN Code: NA 1080

Description: The request cannot be completed while the connection is broken.

User Action: No action required.

--ERROR NA 1520-- Network descriptor list is empty.

Condition Identifier: NAE\$EMPTY_NETWORK_DESC_LIST Code: NA 1520

Description: Network descriptor list is empty.

User Action: No action required.

--INFORMATIVE NA 1550-- Networks are already activated.

Condition Identifier: NAE\$NETWORKS_ALREADY_CONFIGURED Code: NA 1550

Description: The network has previously been configured.

User Action: No action required.

--ERROR NA 1560-- Networks are not activated.

Condition Identifier: NAE\$NETWORKS_NOT_ACTIVATED Code: NA 1560

Description: Networks are not activated. The requested activity cannot be performed.

User Action: No action required.

--ERROR NA 1610-- The network device {text} down notification received from the PP.

Condition Identifier: NAE\$DEVICE_DOWN_VIA_PP Code: NA 1610

Description: The PP driver for the specified network device has determined that the device should be downed. Consult engineering log for specific errors detected.

User Action: Notify customer engineer; dump engineering log.

--ERROR NA 1615-- The reset down threshold has been exceeded for network device {text}.

Condition Identifier: NAE\$DEVICE_RESET_THRESH_EXCEED Code: NA 1615

Description: The number of device resets for the specified network device has exceeded the threshold. The device will be downed.

User Action: Notify the network administrator

--ERROR NA 1635-- The ICA {text} is generating log messages with incorrect statistic codes.

Condition Identifier: NAE\$ICA_STAT_CODE_OUT_OF_RANGE Code: NA 1635

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1640-- Switch to the alternate ICA {text} failed.

Condition Identifier: NAE\$ICA_SWITCH_FAILED Code: NA 1640

Description: The ICA peripheral processor (PP) or channel could not be reserved to activate the secondary ICA.

User Action: Check the PP assignments. Some equipment may have to be turned off to free up a PP.

--ERROR NA 1645-- ICA {text}: The PP and intranet layer mgmt task are out of synch.

Condition Identifier: NAE\$ICA_UNEXPECTED_END_OF_DUMP Code: NA 1645

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1650-- An ill formatted log message was received from {text}.

Condition Identifier: NAE\$ILL_FORMATTED_ICA_LOG_MSG Code: NA 1650

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1655-- Unable to lock table in mtr mode.

Condition Identifier: NAE\$MTR_UNABLE_TO_LOCK_TABLE Code: NA 1655

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1660-- The reset down timer for network device {text} has expired.

Condition Identifier: NAE\$NET_DEVICE_RESET_TIMEOUT Code: NA 1660

Description: The specified network device has failed to become available for use within the timeout period.
The device will be downed.

User Action: Notify customer engineer; dump engineering log.

--ERROR NA 1680-- State change has already been initiated for the given element.

Condition Identifier: NAE\$STATE_CHANGE_IN_PROGRESS Code: NA 1680

Description: A state change has been initiated by the system for the given element.

User Action: No action required.

--ERROR NA 1687-- Network device {text} in unexpected state.

Condition Identifier: NAE\$UNEXPECTED_DEVICE_STATE Code: NA 1687

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1690-- Unexpected {text} response from network PP.

Condition Identifier: NAE\$UNEXPECTED_PP_RESPONSE Code: NA 1690

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 1693-- The element {text} is not present in the network configuration.

Condition Identifier: NAE\$UNKNOWN_ELEMENT Code: NA 1693

Description: The specified element name is not known to NAM/VE.

User Action: Try again with the correct element name.

--ERROR NA 2015-- Title/address already registered.

Condition Identifier: NAE\$DUPLICATE_REGISTRATION Code: NA 2015

Description: Title/address combination is already registered.

User Action: No action required.

--ERROR NA 2040-- Incorrect password given for title entry.

Condition Identifier: NAE\$INCORRECT_PASSWORD Code: NA 2040

Description: Incorrect password given for title entry.

User Action: Retry request with the correct password.

--ERROR NA 2080-- Title/identifier pair not found in directory.

Condition Identifier: NAE\$TITLE_ID_NOT_FOUND Code: NA 2080

Description: Title/identifier pair not found in directory.

User Action: Retry request with the correct title identifier.

--ERROR NA 2085-- Translation request not active.

Condition Identifier: NAE\$TRANSLATION_REQ_NOT_ACTIVE Code: NA 2085

Description: Translation request is not active. Request has timed out or has been deleted. No translations could be found.

User Action: Translate the title then retry request.

--ERROR NA 2095-- Waiting for distributed title.

Condition Identifier: NAE\$WAIT_FOR_DISTRIBUTED_TITLE Code: NA 2095

Description: No title translation is available for a recurrent translation request.

User Action: Retry later.

--ERROR NA 2110-- DEFINE_BOOT_DEFAULTS command must be the first command on the exception list file.

Condition Identifier: NAE\$DEFAULTS_NOT_SET_FIRST Code: NA 2110

Description: DEFINE_BOOT_DEFAULTS command must be the first command on the exception list file.

User Action: Correct CDCNET exception list file.

--ERROR NA 2115-- DEFINE_BOOT_DEFAULTS command can occur only once on the exception list file.

Condition Identifier: NAE\$DEFAULTS_SET_TWICE Code: NA 2115

Description: DEFINE_BOOT_DEFAULTS command can occur only once on the exception list file.

User Action: Correct CDCNET exception list file.

--ERROR NA 2120-- {text} {text} reset with reset code {text}.

Condition Identifier: NAE\$DI_RESET Code: NA 2120

Description: A specific condition has caused the di to reset.

User Action: Notify site analyst.

--ERROR NA 2130-- Load of DI {text} terminated: excessive resyncs.

Condition Identifier: NAE\$EXCESSIVE_RESYNCS Code: NA 2130

Description: The load of DI has been terminated because of excessive resyncs. This would normally occur because of a hardware problem in the communication path to the DI or in the DI itself.

User Action: If the problem persists, notify hardware maintenance.

--ERROR NA 2205-- Improper name {text} specified in file access request.

Condition Identifier: NAE\$BAD_FILE_ACCESS_FILE_NAME Code: NA 2205

Description: A Device Interface (DI) is attempting to access an unknown file.

User Action: Notify a site analyst.

--ERROR NA 2215-- Module with entry point {text} is not a load module.

Condition Identifier: NAE\$EP_NOT_ON_LOAD_MODULE Code: NA 2215

Description: A DI is attempting to load an entry point that does not exist in the CDCNET library.

User Action: Correct the request that initiated the load request or add the entry point to the CDCNET library.

--ERROR NA 2220-- File type {text} unavailable.

Condition Identifier: NAE\$FILE_TYPE_UNAVAILABLE Code: NA 2220

Description: Network file access is unable to support the specified network file type.

User Action: No action required.

--ERROR NA 2230-- Invalid access mode for file {text}.

Condition Identifier: NAE\$INVALID_ACCESS_MODE Code: NA 2230

Description: Network file access received and rejected a request with an invalid access mode for the specified network file. Internal error.

User Action: Notify site analyst to submit PSR.

--ERROR NA 2232-- Invalid File Access request - {text}.

Condition Identifier: NAE\$INVALID_FILE_ACCESS_REQUEST Code: NA 2232

Description: Network file access received and rejected an invalid request. Internal error.

User Action: Notify site analyst to submit PSR.

--ERROR NA 2248-- Unable to create new file in catalog {text}, file limit reached.

Condition Identifier: NAE\$MAX_FILES_REACHED Code: NA 2248

Description: Unable to create new ICA or DI dump file in catalog, file limit reached.

User Action: Archive, delete, or dump any unnecessary ICA or DI dump files; or increase the limit on dump files if more dumps are desired.

--ERROR NA 2255-- Module {text} is not a load module.

Condition Identifier: NAE\$MODULE_NOT_A_LOAD_MODULE Code: NA 2255

Description: A DI is attempting to load a module that cannot be loaded.

User Action: Correct the DI request that initiated the load request or add the module to the DI library.

--ERROR NA 2258-- Module {text} is not a program description.

Condition Identifier: NAE\$MODULE_NOT_PROGRAM_DESC Code: NA 2258

Description: The module specified is not a program description on the CDCNET product library.

User Action: Add the appropriate program description to the CDCNET library or reinstall the CDCNET software.

--ERROR NA 2270-- Procedure {text} not on library {file}.

Condition Identifier: NAE\$PROCEDURE_NOT_ON_LIBRARY Code: NA 2270

Description: A CDCNET terminal or user procedure was not found on the specified library.

User Action: Add the procedure to the library or correct the procedure request to specify a valid procedure name.

--ERROR NA 2275-- Record size overflow on file {text}.

Condition Identifier: NAE\$RECORD_SIZE_OVERFLOW Code: NA 2275

Description: Internal error.

User Action: Notify site analyst to submit PSR.

--ERROR NA 2290-- Write beyond file limit.

Condition Identifier: NAE\$WRITE_BEYOND_FILE_LIMIT Code: NA 2290

Description: Internal error.

User Action: Notify site analyst to submit PSR.

--ERROR NA 2315-- Group {text} specified twice, group names must be unique.

Condition Identifier: NAE\$DUPLICATE_GROUP Code: NA 2315

Description: Group specified twice, group names must be unique.

User Action: Specify a unique group name.

--ERROR NA 2338-- Log group name {text} is invalid, must be an SCL name.

Condition Identifier: NAE\$INVALID_LOG_GROUP_NAME Code: NA 2338

Description: Log groups are specified via SCL names, but the value specified is not a valid SCL name.

User Action: Correct and retry.

--ERROR NA 2340-- Log group priority {text} is invalid, must be an integer between 1 and 255.

Condition Identifier: NAE\$INVALID_LOG_GROUP_PRIORITY Code: NA 2340

Description: Log group priority invalid, must be an integer in the range 1..255.

User Action: Correct the priority parameter to specify a valid value.

--INFORMATIVE NA 2350-- Network Log File terminated - Submitting Log processing job.

Condition Identifier: NAE\$LOG_FILE_TERMINATED Code: NA 2350

Description: Log File terminated - Submitting Log processing job.

User Action: No action required.

--ERROR NA 2360-- Unable to create new cycle of file {text}, cycle limit reached.

Condition Identifier: NAE\$MAX_CYCLES_REACHED Code: NA 2360

Description: Unable to create new cycle of the network log file, cycle limit reached.

User Action: Archive, process, delete, or dump any unnecessary log file cycles; or increase the limit on log file cycles.

--WARNING NA 2386-- Unable to create a new cycle of the Network Log file.

Condition Identifier: NAE\$UNABLE_TO_CREATE_LOG_FILE Code: NA 2386

Description: Unable to create a new cycle of the network log file.

User Action: Archive, process, delete, or dump cycle 999 of network log file. If cycle 999 does not exist, notify site analyst.

--ERROR NA 2390-- Unable to process Network Log File.

Condition Identifier: NAE\$UNABLE_TO_PROCESS_LOG_FILE Code: NA 2390

Description: An error occurred while attempting to attach and submit the Log Processing Job.

User Action: No action required.

--ERROR NA 3005-- An unexpected ACCEPT CONNECTION has been received.

Condition Identifier: NAE\$ACCEPT_NOT_PENDING Code: NA 3005

Description: The connection has either already been established or it has been terminated.

User Action: Check the network file's CONNECTION_STATE attribute. If the connection is established, the connection has already been accepted. Otherwise, the connection has been terminated and cannot be established.

--ERROR NA 3007-- {text} version {text} does not match expected version {text}.

Condition Identifier: NAE\$ACCT_VERSION_MISMATCH Code: NA 3007

Description: An internal error has occurred.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 3010-- Improper network address kind for specified protocol.

Condition Identifier: NAE\$ADDRESS_PROTOCOL_MISMATCH Code: NA 3010

Description: Address protocol mismatch.

User Action: No action required.

--ERROR NA 3020-- Job is already attached to server {text}.

Condition Identifier: NAE\$APPL_ALREADY_ATTACHED Code: NA 3020

Description: The server job is already attached.

User Action: No action required.

--ERROR NA 3030-- The number of jobs attached to server {text} are sufficient to service the maximum number of connections allowed.

Condition Identifier: NAE\$APPLICATION_ATTACH_LIMIT Code: NA 3030

Description: The maximum number of server jobs are attached.

User Action: Retry request later.

--ERROR NA 3040-- Application {text} is inactive.

Condition Identifier: NAE\$APPLICATION_INACTIVE Code: NA 3040

Description: The application is already inactive.

User Action: Retry request when the application is active.

--ERROR NA 3050-- The number of connections for the client {text} is at the maximum allowed.

Condition Identifier: NAE\$APPLICATION_MAX_CONN_LIMIT Code: NA 3050

Description: Request cannot be completed. The maximum connection limit has been reached.

User Action: No action required.

--ERROR NA 3080-- The protocol on the {text} request does not match the expected protocol.

Condition Identifier: NAE\$CLIENT_PROTOCOL_MISMATCH Code: NA 3080

Description: The protocol specified on the request does not match the client application's protocol attributes.

User Action: Check the client application's protocol attribute and retry the request with the correct protocol.

--ERROR NA 3100-- Connection not established.

Condition Identifier: NAE\$CONNECTION_NOT_ESTABLISHED Code: NA 3100

Description: The request cannot be completed. A connection has not been established.

User Action: No action required.

--ERROR NA 3130-- Connection has been terminated.

Condition Identifier: NAE\$CONNECTION_TERMINATED Code: NA 3130

Description: The connection with the communicating peer has been broken. The break was either initiated by the peer or caused by a network failure.

User Action: Reestablish the connection and retry the request.

--ERROR NA 3140-- Data transfer operation terminated because timer expired.

Condition Identifier: NAE\$DATA_TRANSFER_TIMEOUT Code: NA 3140

Description: The request did not complete in the time allowed. The connection may be left in an indeterminate state.

User Action: Synchronize the connection.

--ERROR NA 3150-- {text}: Directory data may not contain more than 32 bytes.

Condition Identifier: NAE\$DIRECTORY_DATA_TOO_LARGE Code: NA 3150

Description: Network directory data is limited to at most 32 bytes.

User Action: Correct request and retry.

--INFORMATIVE NA 3155-- All translations have been delivered.

Condition Identifier: NAE\$DIRECTORY_SEARCH_COMPLETE Code: NA 3155

Description: The directory has completed the requested search. All translations have been delivered.

User Action: Not applicable.

--ERROR NA 3160-- {text}: Parameter {text} has more than one array element with selector {text}.

Condition Identifier: NAE\$DUPLICATE_SELECTOR Code: NA 3160

Description: An attribute is specified more than once in an array of values.

User Action: Correct and retry.

--ERROR NA 3170-- {text}: Title {text} already defined for server {text}.

Condition Identifier: NAE\$DUPLICATE_TITLE Code: NA 3170

Description: The title was specified more than once.

User Action: No action required.

--ERROR NA 3180-- {text}: Title pattern {text} already defined for server {text}.

Condition Identifier: NAE\$DUPLICATE_TITLE_PATTERN Code: NA 3180

Description: The specified title pattern is already in the list of title patterns for the server.

User Action: Not applicable.

--ERROR NA 3320-- The SEND DATA parameter QUALIFIED DATA must remain consistent for an incomplete data message.

Condition Identifier: NAE\$INCONSISTENT_QUALIFIED_DATA Code: NA 3320

Description: Qualified data must be consistent for an incomplete data message.

User Action: Retry the request with qualified data consistent with the previous SEND_DATA request.

--ERROR NA 3330-- Insufficient network resources are available for the {text} request to complete.

Condition Identifier: NAE\$INSUFFICIENT_RESOURCES Code: NA 3330

Description: Available system resources are temporarily busy.

User Action: Retry later.

--ERROR NA 3345-- The CONNECT_DATA attribute may not be changed for an established connection.

Condition Identifier: NAE\$INVALID_CONNECT_DATA_CHANGE Code: NA 3345

Description: The connection has already been accepted.

User Action: Retry the request without the CONNECT_DATA attribute.

--ERROR NA 3350-- {text}: Directory priority must be greater than 0.

Condition Identifier: NAE\$INVALID_DIRECTORY_PRIORITY Code: NA 3350

Description: Directory priority was specified as a negative number.

User Action: Retry with valid priority value.

--ERROR NA 3355-- Invalid search_identifier on {text} request.

Condition Identifier: NAE\$INVALID_DIRECTORY_SEARCH_ID Code: NA 3355

Description: A directory request was made with an unknown directory search identifier. The directory search identifier may be unknown because the search request has expired, or because the identifier does not match the identifier returned by NAPS\$BEGIN_DIRECTORY_SEARCH.

User Action: Verify the program logic and retry the search.

--ERROR NA 3360-- The eoi message size exceeds the maximum allowed value of {text}.

Condition Identifier: NAE\$INVALID_EOI_MESSAGE_SIZE Code: NA 3360

Description: The EOI message size attribute exceeds the accepted maximum.

User Action: Initialize the EOI message size attribute, then retry the request.

--ERROR NA 3370-- {text}: Parameter {text} has an invalid selector in array element {text}.

Condition Identifier: NAE\$INVALID_SELECTOR Code: NA 3370

Description: An invalid selector value is specified in the indicated entry of an array of values.

User Action: Correct the entry and retry.

--ERROR NA 3500-- Maximum connections allowed for server {text} have been acquired.

Condition Identifier: NAE\$MAX_CONNECTIONS_ACQUIRED Code: NA 3500

Description: The maximum number of connections have been acquired.

User Action: No action required.

--ERROR NA 3520-- Specified maximum connections value does not match server application definition.

Condition Identifier: NAE\$MAX_CONNECTIONS_MISMATCH Code: NA 3520

Description: There is a mismatch in maximum number of connections per server job value.

User Action: Retry the request specifying the correct value for maximum connections.

--ERROR NA 3540-- Data length exceeds {text}.

Condition Identifier: NAE\$MAX_DATA_LENGTH_EXCEEDED Code: NA 3540

Description: Data length exceeds the length allowed by the request.

User Action: Reduce the amount of data specified on the request.

--ERROR NA 3570-- Multiple attempts to wait on {text}.

Condition Identifier: NAE\$MULTIPLE_WAITS_ATTEMPTED Code: NA 3570

Description: Two tasks are attempting to wait for the same operation. One of the requests has been rejected.

User Action: Retry request after the waiting task has completed.

--ERROR NA 3610-- Network is inactive.

Condition Identifier: NAE\$NETWORK_INACTIVE Code: NA 3610

Description: NAM/VE is not active.

User Action: Retry the request later after NAM/VE has been activated.

--ERROR NA 3630-- No connection is available for server {text}.

Condition Identifier: NAE\$NO_CONNECTION_AVAILABLE Code: NA 3630

Description: No connections are available.

User Action: Retry later.

--ERROR NA 3640-- No input data is available.

Condition Identifier: NAE\$NO_DATA_AVAILABLE Code: NA 3640

Description: No data is available to complete the request.

User Action: Retry later.

--ERROR NA 3650-- The server application has not responded to a connection request.

Condition Identifier: NAE\$NO_SERVER_RESPONSE Code: NA 3650

Description: No server response received.

User Action: Consult operations to determine the status of the server.

--ERROR NA 3660-- No switch offer is pending on connection {text}.

Condition Identifier: NAE\$NO_SWITCH_OFFER_PENDING Code: NA 3660

Description: The request may only be made while a connection is being switched.

User Action: No action required.

--ERROR NA 3670-- No connection switch has been offered.

Condition Identifier: NAE\$NO_SWITCH_OFFERED Code: NA 3670

Description: No switched connections are available.

User Action: No action required.

--ERROR NA 3675-- {text}: No {text} server title matches title pattern {text}.

Condition Identifier: NAE\$NO_TITLE_MATCH Code: NA 3675

Description: A request to delete server managed titles has failed because no titles match the pattern specified.

User Action: Verify the title pattern specified and compare it to the current list of server managed titles.

--INFORMATIVE NA 3680-- No translations are currently available for delivery.

Condition Identifier: NAE\$NO_TRANSLATION_AVAILABLE Code: NA 3680

Description: No translations are currently available for the specified directory search. The search is still in progress.

User Action: Not applicable.

--ERROR NA 3720-- A receive data operation is already in progress.

Condition Identifier: NAE\$RECEIVE_OUTSTANDING Code: NA 3720

Description: No receive requests may be outstanding while attempting an asynchronous receive request.

User Action: Retry later.

--ERROR NA 3730-- Interrupt data cannot exceed {text} bytes and cannot be smaller than {text} bytes.

Condition Identifier: NAE\$SE_INTERRUPT_LENGTH_ERROR Code: NA 3730

Description: The interrupt data length parameter is incorrect.

User Action: Reinitialize the INTERRUPT_DATA_LENGTH parameter and retry the request.

--ERROR NA 3735-- Synchronization of output data is not in progress. A nap\$se_synchronize_confirm request is not allowed.

Condition Identifier: NAE\$SE_NO_SYNCH_IN_PROGRESS Code: NA 3735

Description: No synchronize requests are active.

User Action: No action required.

--ERROR NA 3745-- The user's peer must respond to the previously issued synchronize before another synchronize request can be processed.

Condition Identifier: NAE\$SE_SYNCH_CONFIRM_PENDING Code: NA 3745

Description: The peer application has not confirmed an outstanding synchronization.

User Action: Wait for the peer application's synchronization confirm, then retry the request.

--ERROR NA 3755-- A SYNCHRONIZE CONFIRM is required before the request can be accepted.

Condition Identifier: NAE\$SE_SYNCHRONIZE_IN_PROGRESS Code: NA 3755

Description: The peer application has synchronized the network connection.

User Action: Confirm the synchronization, then retry the request.

--ERROR NA 3760-- Synchronize data cannot exceed {text} bytes and cannot be smaller than {text} bytes.

Condition Identifier: NAE\$SE_SYNCHRONIZE_LENGTH_ERROR Code: NA 3760

Description: The SYNCHRONIZE_DATA_LENGTH parameter is incorrect.

User Action: Reinitialize the SYNCHRONIZE_DATA_LENGTH parameter and retry the request.

--ERROR NA 3770-- The synchronize direction specified is unknown.

Condition Identifier: NAE\$SE_UNKNOWN_SYNCH_DIRECTION Code: NA 3770

Description: The SYNCHRONIZE_DIRECTION parameter is incorrect.

User Action: Initialize the SYNCHRONIZE_DIRECTION parameter and retry the request.

--ERROR NA 3780-- A send data operation is already in progress.

Condition Identifier: NAE\$SEND_OUTSTANDING Code: NA 3780

Description: No send requests may be outstanding while attempting an asynchronous send request.

User Action: Retry later.

--ERROR NA 3800-- Job is not attached to server {text}.

Condition Identifier: NAE\$SERVER_NOT_ATTACHED Code: NA 3800

Description: The server job is not attached.

User Action: Retry the request after attaching the job to the server.

--ERROR NA 3820-- The remote server application did not respond to the connection request for {text}.

Condition Identifier: NAE\$SERVER_RESPONSE_TIMEOUT Code: NA 3820

Description: The server response has timed out.

User Action: Consult operations to determine the status of the server.

--ERROR NA 3840-- Connection {text} has been switched to another job.

Condition Identifier: NAE\$SWITCH_OFFER_ACCEPTED Code: NA 3840

Description: The request cannot be completed. The switch offer has been accepted.

User Action: No action required.

--ERROR NA 3845-- Switch offer for connection {text} was not accepted.

Condition Identifier: NAE\$SWITCH_OFFER_NOT_ACCEPTED Code: NA 3845

Description: The switch offer has not been accepted.

User Action: Cancel the switch offer.

--ERROR NA 3850-- Switch offer is pending on connection {text}.

Condition Identifier: NAE\$SWITCH_OFFER_PENDING Code: NA 3850

Description: The request cannot be completed while a switch offer is pending.

User Action: No action required.

--ERROR NA 3870-- {text}: Title {text} is not authorized for server {text}.

Condition Identifier: NAE\$TITLE_NOT_AUTHORIZED Code: NA 3870

Description: A server application has attempted to add a server managed title that does not match an authorized title pattern.

User Action: Verify the list of authorized title patterns.

--ERROR NA 3875-- {text}: Title length ({text}) exceeds maximum allowed (255).

Condition Identifier: NAE\$TITLE_TOO_LONG Code: NA 3875

Description: A network directory title is limited to 255 characters.

User Action: Correct the request and retry.

--ERROR NA 3880-- {text}: A title must contain at least 1 character.

Condition Identifier: NAE\$TITLE_TOO_SHORT Code: NA 3880

Description: An empty string was specified as a title.

User Action: Retry with a valid title.

--ERROR NA 3890-- {text}: Title pattern length ({text}) exceeds maximum allowed (255).

Condition Identifier: NAE\$TITLE_PATTERN_TOO_LONG Code: NA 3890

Description: A title pattern is limited to 255 characters.

User Action: Retry with a valid title pattern.

--ERROR NA 3900-- {text}: A title pattern must contain at least 1 character.

Condition Identifier: NAE\$TITLE_PATTERN_TOO_SHORT Code: NA 3900

Description: An empty string was specified as a title pattern.

User Action: Retry with a valid title pattern.

--ERROR NA 3920-- Peer application has performed a connection operation other than data transfer.

Condition Identifier: NAE\$UNEXPECTED_PEER_OPERATION Code: NA 3920

Description: An unexpected peer operation has been received.

User Action: No action required.

--ERROR NA 3940-- The address kind specified is unknown.

Condition Identifier: NAE\$UNKNOWN_ADDRESS_KIND Code: NA 3940

Description: The address parameter is incorrect.

User Action: Reinitialize the address parameter and retry the request.

--ERROR NA 3950-- Application {text} has not been defined.

Condition Identifier: NAE\$UNKNOWN_APPLICATION Code: NA 3950

Description: No application by the name specified was found.

User Action: Retry request with the correct application name or use the keyword ALL to list the information for all applications.

--ERROR NA 3960-- An attribute specified {text} is unknown.

Condition Identifier: NAE\$UNKNOWN_ATTRIBUTE Code: NA 3960

Description: An attribute specified is unknown.

User Action: Initialize the attribute parameter, then retry the request.

--ERROR NA 3970-- Unknown protocol.

Condition Identifier: NAE\$UNKNOWN_PROTOCOL Code: NA 3970

Description: A data unit has been received specifying an unknown protocol.

User Action: Retry the request with the correct protocol.

--ERROR NA 3995-- Only address kinds of nac\$osi_transport_address and nac\$internet_address are supported.

Condition Identifier: NAE\$UNSUPPORTED_ADDRESS Code: NA 3995

Description: The address kind field is invalid.

User Action: Retry the request with an address kind of either NAC\$OSI_TRANSPORT_ADDRESS or NAC\$INTERNET_ADDRESS.

--ERROR NA 4002-- Application {text} has active connections.

Condition Identifier: NAE\$ACTIVE_CONNECTIONS Code: NA 4002

Description: The server cannot be deleted because it has active connections.

User Action: Retry the request after all active connections have terminated.

--ERROR NA 4005-- Application {text} is active.

Condition Identifier: NAE\$APPLICATION_ACTIVE Code: NA 4005

Description: The application is already active.

User Action: Retry the request after deactivating the application.

--ERROR NA 4010-- Application {text} is already active.

Condition Identifier: NAE\$APPLICATION_ALREADY_ACTIVE Code: NA 4010

Description: The application is already active.

User Action: No action required.

--ERROR NA 4015-- {text} Application is already defined.

Condition Identifier: NAE\$APPLICATION_ALREADY_DEFINED Code: NA 4015

Description: The application is already defined.

User Action: Determine if the current definition is correct. Redefine the application or select a new application name if necessary.

--ERROR NA 4020-- Application {text} is already inactive.

Condition Identifier: NAE\$APPLICATION_ALREADY_INACTIV Code: NA 4020

Description: The application is already inactive.

User Action: No action required.

--ERROR NA 4025-- {text} The definitions on application file are incorrect.

Condition Identifier: NAE\$APPLICATION_FILE_ERROR Code: NA 4025

Description: The definitions on application file are incorrect.

User Action: Notify site analyst. Submit a PSR and a display of all attached cycles of the application definition file.

--ERROR NA 4030-- Application file mismatch.

Condition Identifier: NAE\$APPLICATION_FILE_MISMATCH Code: NA 4030

Description: The application definitions on this file do not match the installed application definitions.

User Action: Notify your site analyst. The highest cycle of the application definitions file may have been deleted.

--ERROR NA 4035-- {text} Application definitions file is not open.

Condition Identifier: NAE\$APPLICATION_FILE_NOT_OPEN Code: NA 4035

Description: The application definitions file is not open.

User Action: Internal error. Submit a psr.

--ERROR NA 4038-- An error occurred while changing the {text} application and also while restoring the application to it's original state. See the job log for details.

Condition Identifier: NAE\$APPLICATION_NOT_RESTORED Code: NA 4038

Description: Errors occurred while changing the application's attributes and also when the application's attributes were being restored to their original values. The application was deactivated because of the errors. Inspect the job log for the exact error descriptions.

User Action: Notify your site analyst. The application's attributes may be able to be changed now that the application is inactive.

--ERROR NA 4050-- Reserved SAP {text} is already in use.

Condition Identifier: NAE\$CONFLICTING_RESERVED_SAP Code: NA 4050

Description: The specified reserved application identifier is already in use.

User Action: Resolve the conflict in application identifier usage.

--WARNING NA 4055-- Cycle 1 of application definitions file was present and was purged.

Condition Identifier: NAE\$CYCLE_1_PRESENT Code: NA 4055

Description: Cycle 1 of application definitions file is present, which means the previous attempt to update the network application definitions failed.

User Action: Notify site analyst.

--ERROR NA 4095-- The application file version {text} is incorrect.

Condition Identifier: NAE\$INCORRECT_APPL_FILE_VERSION Code: NA 4095

Description: The definitions on the application file are incorrect.

User Action: Notify your site analyst.

--ERROR NA 4100-- Attached server jobs are at maximum connections limit.

Condition Identifier: NAE\$INSUFFICIENT_ATTACHED_JOBS Code: NA 4100

Description: The attached server jobs have reached their maximum connections limit.

User Action: Start another instance of the server to process additional connections.

--ERROR NA 4150-- No server job is attached to {text}.

Condition Identifier: NAE\$NO_SERVER_JOB_ATTACHED Code: NA 4150

Description: A program interface request has been made by a self-initiated server application when no server jobs are currently attached.

User Action: Issue an attach server request and retry the rejected request.

--ERROR NA 4250-- Server jobs attached to {text}.

Condition Identifier: NAE\$SERVER_JOBS_ATTACHED Code: NA 4250

Description: The server has jobs attached.

User Action: Retry the request after all server jobs are detached.

--WARNING NA 4260-- Maximum connections limit for {text} has been set to the maximum allowed value of {text}.

Condition Identifier: NAE\$LIMIT_MAX_CONNECTIONS Code: NA 4260

Description: The maximum connections limit specified for the given application exceeds the maximum allowed value. It is set to the maximum allowed value by the system.

User Action: Note the maximum allowed value and correct the application definition procedure to specify the correct value.

--ERROR NA 4400-- Unable to activate application {text} during NAMVE initialization.

Condition Identifier: NAE\$UNABLE_TO_ACTIVATE_APPL Code: NA 4400

Description: An error was encountered while installing this application during NAMVE initialization. Inspect the job log for the exact error description.

User Action: Notify your site analyst. The application may need to be deleted and redefined.

--ERROR NA 4405-- An even number of digits are required. {text} digits were specified.

Condition Identifier: NAE\$EVEN_CHAR_COUNT_REQUIRED Code: NA 4405

Description: The parameter in error must be specified as a string containing an even number of hexadecimal digits. Use an odd number of digits were specified.

User Action: Correct and retry.

--ERROR NA 4410-- Only the numbers 0 through 9 and the letters A through F are allowed. {text} was specified.

Condition Identifier: NAE\$INVALID_HEX_DIGIT Code: NA 4410

Description: The parameter in error must be specified as a string containing an even number of hexadecimal digits. A character that is not a valid hexadecimal digit was specified.

User Action: Correct and retry.

--ERROR NA 4415-- The title identifier {text} is not present in the current list of registered non-CDNA titles.

Condition Identifier: NAE\$UNKNOWN_IDENTIFIER Code: NA 4415

Description: The specified title identifier is not associated with any title currently registered.

User Action: Use the DISPLAY_OSI_ADDRESSES command to list all titles currently registered with their identifiers. Reenter the rejected command with the correct identifier.

--ERROR NA 4420-- The title {text} is not present in the current list of registered OSI titles.

Condition Identifier: NAE\$UNKNOWN_OSI_TITLE Code: NA 4420

Description: The specified title was not found in the current list of registered titles.

User Action: Use the DISPLAY_OSI_ADDRESSES command with the keyword ALL specified for the title parameter to display all registered titles.

--ERROR NA 4425-- No OSI titles are registered in the Directory.

Condition Identifier: NAE\$EMPTY_OSI_TITLE_LIST Code: NA 4425

Description: A request to display a title was made when no titles are currently registered.

User Action: None.

--ERROR NA 4520-- The value ALL is not allowed as the name of an application.

Condition Identifier: NAE\$ALL_NOT_VALID_APPL_NAME Code: NA 4520

Description: The value ALL is not allowed as the name of an application.

User Action: Retry the request specifying a value other than ALL.

--ERROR NA 4525-- No changes to the {text} definition were entered.

Condition Identifier: NAE\$APPL_DEFINITION_NOT_CHANGED Code: NA 4525

Description: The change client or change server subutility was exited without entering any changes to the application's definition.

User Action: No action required.

--ERROR NA 4530-- An application identifier can not be changed to variable if it's current value is variable.

Condition Identifier: NAE\$APPLICATION_ID_NOT_CHANGED Code: NA 4530

Description: The value of the application identifier attribute is already variable.

User Action: No action required.

--ERROR NA 4535-- No {text} applications are defined.

Condition Identifier: NAE\$APPLICATION_NOT_DEFINED Code: NA 4535

Description: No applications of this type are defined.

User Action: No action required.

--ERROR NA 4540-- The specified client address is not in the current list of valid addresses.

Condition Identifier: NAE\$CLIENT_ADDRESS_NOT_IN_LIST Code: NA 4540

Description: The specified client address is not in the current list of valid addresses.

User Action: No action required.

--ERROR NA 4550-- The {text} command is not allowed when {text}.

Condition Identifier: NAE\$COMMAND_NOT_ALLOWED Code: NA 4550

Description: The specified command is not allowed by the MANAGE_NETWORK_APPLICATIONS utility under the specified circumstances.

User Action: Complete the current activity and retry the request.

--ERROR NA 4570-- {text} is not a valid client attribute name.

Condition Identifier: NAE\$INVALID_CLIENT_ATTRIBUTE Code: NA 4570

Description: An invalid client attribute was specified.

User Action: Retry request with valid client attributes.

--ERROR NA 4575-- {text} is not a valid server attribute name.

Condition Identifier: NAE\$INVALID_SERVER_ATTRIBUTE Code: NA 4575

Description: An invalid server attribute was specified.

User Action: Retry request with valid server attributes.

--ERROR NA 4580-- User not authorized to invoke {text}.

Condition Identifier: NAE\$INVALID_USER Code: NA 4580

Description: User is not authorized to perform the specified activity.

User Action: The user's family administrator can authorize the user to perform this activity if appropriate.

--INFORMATIVE NA 4590-- Manage_network_applications is being executed while NAM/VE is inactive.

Condition Identifier: NAE\$MANNA_WHEN_NETWORK_INACTIVE Code: NA 4590

Description: The MANAGE_NETWORK_APPLICATIONS utility was entered and NAM/VE is not active.

User Action: No action required.

--ERROR NA 4600-- Server job files must be mass storage files in the current implementation.

Condition Identifier: NAE\$ONLY_DISK_FILES_SUPPORTED Code: NA 4600

Description: Server job files must be mass storage files in the current implementation.

User Action: Retry request specifying a mass storage file for the server job file.

--ERROR NA 4610-- The page width of the output file must be at least {text}.

Condition Identifier: NAE\$PAGE_WIDTH_TOO_SMALL Code: NA 4610

Description: The page width attribute of the file specified as the output file is too small.

User Action: Specify a file for the output file with a page width attribute of at least the value given in the error message.

--ERROR NA 4620-- A title pattern must be specified as a name or a string, not a {text}.

Condition Identifier: NAE\$PATTERN_NOT_NAME_OR_STRING Code: NA 4620

Description: A title pattern value must be specified as a name or a string.

User Action: Retry the request specifying a valid title pattern.

--ERROR NA 4630-- A title value must be specified as a name or a string, not a {text}.

Condition Identifier: NAE\$TITLE_NOT_NAME_OR_STRING Code: NA 4630

Description: A title value must be specified as a name or a string.

User Action: Retry request specifying a valid title.

--WARNING NA 4640-- Title {text} is not in the current list of titles.

Condition Identifier: NAE\$UNKNOWN_TITLE Code: NA 4640

Description: The title is not in the current list of titles.

User Action: No action required.

--WARNING NA 4645-- Title pattern {text} is not in the current list of title patterns.

Condition Identifier: NAE\$UNKNOWN_TITLE_PATTERN Code: NA 4645

Description: The title pattern is not in the current list of titles.

User Action: No action required.

--ERROR NA 4660-- The parameter value {text} is not supported in the current implementation.

Condition Identifier: NAE\$VALUE_NOT_SUPPORTED Code: NA 4660

Description: One of the parameter values is not supported in the current implementation.

User Action: Retry request without the unsupported parameter.

--WARNING NA 4680-- Warnings detected processing MANNA command {text}. See \$ERRORS.

Condition Identifier: NAE\$WARNINGS_PROCESSING_COMMAND Code: NA 4680

Description: A warning was detected processing a MANAGE_NETWORK_APPLICATIONS command. See \$ERRORS.

User Action: See \$ERRORS.

--ERROR NA 5005-- Connection to system {text} lost after CDCNET command sent.

Condition Identifier: NAE\$ACCESS_LOST Code: NA 5005

Description: The connection to the specified system was lost after the CDCNET command was sent.

User Action: Determine the status of the DI system from its status lights or from any network alarms that were generated.

--ERROR NA 5010-- Alarms are already active for alarm group {text}.

Condition Identifier: NAE\$ALARMS_ALREADY_ACTIVE Code: NA 5010

Description: Alarms are already active for an alarm group.

User Action: No action required.

--ERROR NA 5015-- Alarms are not active for alarm group {text}.

Condition Identifier: NAE\$ALARMS_NOT_ACTIVE Code: NA 5015

Description: Alarms are already inactive for an alarm group.

User Action: No action required.

--ERROR NA 5030-- No response received for connect request to system {text}.

Condition Identifier: NAE\$COMMAND_CONNECTION_IGNORED Code: NA 5030

Description: No response received for connect request to system. The DI may be failing or may be in the process of reloading or access to the DI may have been lost.

User Action: The DI may need to be manually reset.

--ERROR NA 5035-- Unable to establish connection to system {text}.

Condition Identifier: NAE\$COMMAND_CONNECTION_REJECTED Code: NA 5035

Description: Unable to establish a connection to the specified system.

User Action: Attempt to establish a connection via a new instance of NETOU. If the connection request still fails, reset the DI.

--ERROR NA 5040-- The last CDCNET command was not sent to system {text}.

Condition Identifier: NAE\$COMMAND_NOT_SENT_TO_SYSTEM Code: NA 5040

Description: The last CDCNET command was not sent to the designated system.

User Action: Specify the name of a system that received the last command.

--ERROR NA 5050-- Errors detected processing NETOU command {text}. See \$ERRORS.

Condition Identifier: NAE\$ERRORS_DURING_COMMAND Code: NA 5050

Description: Errors detected processing NETOU command. See \$ERRORS.

User Action: See \$ERRORS.

--ERROR NA 5090-- {text} is not a valid alarm group name.

Condition Identifier: NAE\$INVALID_ALARM_GROUP Code: NA 5090

Description: An invalid alarm group name was specified. Only CATENET is supported as an alarm group name in this release.

User Action: Specify a valid alarm group name.

--ERROR NA 5110-- No CDCNET commands have been sent.

Condition Identifier: NAE\$NO_COMMAND_SENT Code: NA 5110

Description: No CDCNET commands have been sent.

User Action: Send a CDCNET command before reissuing this request.

--ERROR NA 5145-- No response received from system {text} for the last CDCNET command.

Condition Identifier: NAE\$RESPONSE_NOT_RECEIVED Code: NA 5145

Description: No response received from system for the last CDCNET command.

User Action: No action required.

--ERROR NA 5160-- Connection to system {text} lost before CDCNET command sent.

Condition Identifier: NAE\$SYSTEM_DISCONNECTED Code: NA 5160

Description: The connection to the specified system was lost before the CDCNET command was sent.

User Action: Retry sending the command.

--ERROR NA 5165-- The name parameter is required for {text} when the last CDCNET command was sent to multiple systems.

Condition Identifier: NAE\$SYSTEM_NAME_REQUIRED Code: NA 5165

Description: The name parameter is required for the last CDCNET command sent to multiple systems.

User Action: Specify system name.

--ERROR NA 5180-- The limit of {text} active broadcast commands has been reached.

Condition Identifier: NAE\$TOO_MANY_BROADCAST_COMMANDS Code: NA 5180

Description: The limit of active broadcast commands has been reached.

User Action: Retry later.

--ERROR NA 5190-- Network message contains unknown kind of data.

Condition Identifier: NAE\$UNKNOWN_CDNA_MDU_DATA_KIND Code: NA 5190

Description: Network message contains unknown kind of data.

User Action: Notify system analyst.

--ERROR NA 5195-- System {text} is unknown.

Condition Identifier: NAE\$UNKNOWN_SYSTEM Code: NA 5195

Description: The system name specified is unknown. If a valid name was specified, the DI may be inaccessible or may be in the process of reloading.

User Action: Check whether system name specified is correct. If it is correct, verify that network access to the system is functional and that the specified system is operational.

--ERROR NA 5196-- Address kind {text}, received from system {text}, is not supported.

Condition Identifier: NAE\$UNSUPPORTED_ADDRESS_KIND Code: NA 5196

Description: A system has registered an address that is not supported by NETOV.

User Action: Notify a system analyst.

--INFORMATIVE NA 5510-- Account:

Condition Identifier: NAE\$ACCOUNT_PROMPT Code: NA 5510

Description: Request for account.

User Action: Enter account.

--INFORMATIVE NA 5525-- Family:

Condition Identifier: NAE\$FAMILY_PROMPT Code: NA 5525

Description: Request for family name.

User Action: Enter family name.

--ERROR NA 5535-- You must enter a name.

Condition Identifier: NAE\$IMPROPER_NAME_AT_LOGIN Code: NA 5535

Description: Improper name specified on login.

User Action: Enter correct name.

--INFORMATIVE NA 5540-- Incorrect validation entered.

Condition Identifier: NAE\$INCORRECT_VALIDATION Code: NA 5540

Description: Validation is incorrect.

User Action: Reenter login with correct validation.

--INFORMATIVE NA 5542-- Login command parameter is either missing or invalid.

Condition Identifier: NAE\$LOGIN_PARAMETER_INVALID Code: NA 5542

Description: Either a required login parameter was not specified or an improper name was specified for a parameter.

User Action: Enter the parameters correctly.

--INFORMATIVE NA 5545-- Enter validation for service access.

Condition Identifier: NAE\$LOGIN_BANNER Code: NA 5545

Description: Login banner.

User Action: Enter validation for service access.

--ERROR NA 5550-- Response timeout. Connection rejected.

Condition Identifier: NAE\$LOGIN_TIMEOUT Code: NA 5550

Description: Login time limit reached.

User Action: No action required.

--ERROR NA 5554-- You must enter a {text}.

Condition Identifier: NAE\$LOGIN_VALIDATION_REQUIRED Code: NA 5554

Description: Login validation is required for user access to system.

User Action: Enter username and/or password when prompted.

--INFORMATIVE NA 5560-- {text}

Condition Identifier: NAE\$NULL_MESSAGE Code: NA 5560

Description: Internal error.

User Action: No action required.

--INFORMATIVE NA 5570-- Password:

Condition Identifier: NAE\$PASSWORD_PROMPT Code: NA 5570

Description: Request for password value.

User Action: Enter password.

--INFORMATIVE NA 5572-- Project:

Condition Identifier: NAE\$PROJECT_PROMPT Code: NA 5572

Description: Request for project.

User Action: Enter the project number.

--INFORMATIVE NA 5574-- No more retries allowed. Connection rejected.

Condition Identifier: NAE\$RETRY_LIMIT_AT_LOGIN Code: NA 5574

Description: No more retries allowed. Connection rejected.

User Action: Consult site administrator to determine proper login sequence.

--INFORMATIVE NA 5577-- Please try again.

Condition Identifier: NAE\$RETRY_LOGIN Code: NA 5577

Description: Login failed.

User Action: Reenter login.

--INFORMATIVE NA 5593-- User:

Condition Identifier: NAE\$USER_PROMPT Code: NA 5593

Description: Request for user's name.

User Action: Enter user name.

--FATAL NA 6010-- A NAM/VE initialization error occurred activating the network configuration.

Condition Identifier: NAE\$ACTIVATE_NETWORK_CONFIG Code: NA 6010

Description: An error occurred while attempting to activate network configuration.

User Action: Notify the network administrator.

--FATAL NA 6040-- A NAM/VE initialization error occurred installing applications.

Condition Identifier: NAE\$INSTALL_APPLICATIONS Code: NA 6040

Description: An error occurred while installing network applications.

User Action: Notify the network administrator.

--ERROR NA 6050-- A NAM/VE initialization error occurred in phase {text}.

Condition Identifier: NAE\$INITIALIZATION_ERROR Code: NA 6050

Description: An error occurred while attempting to activate networks.

User Action: Notify the network administrator.

--FATAL NA 6055-- A *FATAL* NAM/VE initialization error occurred in phase {text}.

Condition Identifier: NAE\$INITIALIZATION_FATAL Code: NA 6055

Description: An error occurred while attempting to activate networks.

User Action: Notify the network administrator.

--WARNING NA 6060-- A NAM/VE initialization error occurred.

Condition Identifier: NAE\$INITIALIZATION_WARNING Code: NA 6060

Description: A nonfatal error occurred while attempting to activate networks.

User Action: Notify the network administrator.

--WARNING NA 6080-- The parameter {text} is not supported in the current implementation.

Condition Identifier: NAE\$PARAMETER_NOT_SUPPORTED Code: NA 6080

Description: The parameter is no longer supported; therefore, its value is ignored.

User Action: None.

--ERROR NA 6201-- No network device is currently available.

Condition Identifier: NAE\$SM_NO_DEVICE_CONFIGURED Code: NA 6201

Description: No device is currently configured with the OSI stack.

User Action: Configure the communications device(s) with the OSI stack and retry.

--ERROR NA 6202-- No route is known to the given destination address.

Condition Identifier: NAE\$SM_ROUTE_UNKNOWN Code: NA 6202

Description: There is no available route to the given destination address.

User Action: Notify the network administrator.

--ERROR NA 6203-- The network devices are either in flow control or are being initialized.

Condition Identifier: NAE\$SM_DEVICES_INACCESSIBLE Code: NA 6203

Description: The configured devices are unavailable due to a resource constraint or the devices have been reset.

User Action: Notify a site analyst.

--ERROR NA 6204-- The system management connection disconnected by the peer with a reason code of {text}.

Condition Identifier: NAE\$SM_PEER_DISCONNECT Code: NA 6204

Description: The communications device that is initializing, detected an error and disconnected the channel connection with the system management access agent in the host.

User Action: Notify a site analyst.

--ERROR NA 6205-- The network device {text} rejected the host network address.

Condition Identifier: NAE\$SM_DUP_HOST_ADDRESS_REJECT Code: NA 6205

Description:

User Action: Not applicable.

--ERROR NA 6301-- The OSI network layer sap {text} is not open.

Condition Identifier: NAE\$NA_SAP_NOT_OPEN Code: NA 6301

Description: The specified network layer sap is not open.

User Action: Notify a site analyst.

--ERROR NA 6302-- The OSI network layer sap {text} closed by the peer via {text}.

Condition Identifier: NAE\$NA_PEER_DISCONNECT Code: NA 6302

Description: The specified network layer sap was closed by the peer in the communications device. The device may have been reset.

User Action: Notify a site analyst.

--ERROR NA 6303-- The OSI network layer sap {text} is already open.

Condition Identifier: NAE\$NA_SAP_ALREADY_OPEN Code: NA 6303

Description: The specified network layer sap is already open.

User Action: Notify a site analyst.

--ERROR NA 6304-- The OSI network layer sap {text} is not open in the device.

Condition Identifier: NA\$NA_DEVICE_SAP_NOT_OPEN Code: NA 6304

Description: The specified network layer sap is not open in the device. The communications device may have been reset.

User Action: Notify a site analyst.

--ERROR NA 7010-- Accept data length cannot exceed {text} bytes.

Condition Identifier: NA\$TA_ACCEPT_DATA_LENGTH_ERROR Code: NA 7010

Description: Internal error.

User Action: Notify the site analyst to submit a PSR.

--ERROR NA 7020-- An accept connection request is not pending.

Condition Identifier: NA\$TA_ACCEPT_CONN_NOT_PENDING Code: NA 7020

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7030-- Connect data length cannot exceed {text} bytes.

Condition Identifier: NA\$TA_CONNECT_DATA_LEN_ERROR Code: NA 7030

Description: Internal error.E

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7040-- Connection not established.

Condition Identifier: NA\$TA_CONNECT_NOT_ESTABLISHED Code: NA 7040

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7050-- The connection has been terminated.

Condition Identifier: NA\$TA_CONNECTION_TERMINATED Code: NA 7050

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7060-- Data must be greater than {text} bytes.

Condition Identifier: NA\$TA_DATA_LENGTH_ERROR Code: NA 7060

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7070-- Disconnect data length cannot exceed {text} bytes.

Condition Identifier: NA\$TA_DISCONNECT_DATA_LEN_ERR Code: NA 7070

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7080-- Expedited data cannot exceed {text} bytes and cannot be smaller than {text} bytes.

Condition Identifier: NA\$TA_EXPEDITED_LENGTH_ERROR Code: NA 7080

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7090-- The expedited data request cannot be accepted because the expedited data option was disabled during connection establishment.

Condition Identifier: NA\$TA_EXPEDITED_NOT_SUPPORTED Code: NA 7090

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7100-- The connection has reached the expedited request limit. The communicating peer must confirm the outstanding expedited requests.

Condition Identifier: NAE\$TA_EXPEDITED_REQUEST_LIMIT Code: NA 7100

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NA 7110-- The only acceptable nlt\$ta_aggregate kinds are nlt\$ta_data_event and nlt\$ta_expedited_data_event.

Condition Identifier: NAE\$TA_IMPROPER_AGGREGATE_KIND Code: NA 7110

Description: Internal error.

User Action: Notify a site analyst to submit a PSR.

--ERROR NF 0-- no user directives supplied for job file.

Condition Identifier: NFE\$JOB_FILE_ERROR Code: NF 0

Description: Batch transfer services received no user directives with which to construct the remote user job.

User Action: Retry Manage_Remote_Files or implicit file transfer. If error persists, notify Site Analyst.

--ERROR NF 2-- The {text} parameter is required for the {text} command.

Condition Identifier: NFE\$REQUIRED_PARAMETER_MISSING Code: NF 2

Description: Batch Transfer Services detected a missing parameter which was designated as required by the RHF A - A Protocol Specification.

User Action: Notify your site analyst.

--ERROR NF 3-- invalid command code.

Condition Identifier: NFE\$INVALID_COMMAND_CODE Code: NF 3

Description: Batch Transfer Services received a command whose code was improper for the current state of the protocol.

User Action: Transfer can be retried. If condition repeats, notify site analyst.

--ERROR NF 4-- Service is not established with location {text}.

Condition Identifier: NFE\$SERVICE_NOT_FOUND Code: NF 4

Description: PTFS has not registered a title for the location requested.

User Action: Check the family name for implicit file accesses and the location parameter on the manage_remote_files command and correct if necessary.

--ERROR NF 5-- unknown data format declared.

Condition Identifier: NFE\$UNKNOWN_DATA_FORMAT Code: NF 5

Description: Batch transfer services received a data declaration parameter value not recognized as one specified by the RHF A-A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 6-- parameter code {text} is unknown to the A to A protocol

Condition Identifier: NFE\$UNRECOGNIZABLE_PARAMETER Code: NF 6

Description: Batch transfer services received a parameter not recognized as one specified by the RHF A-A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 7-- qualifier found is unknown to the A to A protocol

Condition Identifier: NFE\$UNRECOGNIZABLE_QUALIFIER Code: NF 7

Description: Batch transfer services received a qualifier for a parameter not recognized as one specified by the RHF A-A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 9-- {text} is an illegal qualifier for the {text} parameter on the {text} command.

Condition Identifier: NFE\$QUALIFIER_ERROR Code: NF 9

Description: Batch transfer services received a qualifier not permitted by the RHF A-A protocol specification for the received parameter.

User Action: Notify your site analyst.

--ERROR NF 10-- {text} is an prohibited parameter for the {text} command.

Condition Identifier: NFE\$PROHIBITED_PARAMETER Code: NF 10

Description: Batch transfer services received a parameter that is prohibited by the RHF A - A Protocol Specification for the command being processed.

User Action: Notify your site analyst.

--ERROR NF 11-- The {text} command parameter of {text} does not correspond to the size limits of the protocol.

Condition Identifier: NFE\$PARAMETER_LENGTH_ERROR Code: NF 11

Description: Batch transfer services received a protocol command that contained a parameter whose size was not within the limits specified by the RHF A - A protocol.

User Action: Notify your site analyst.

--ERROR NF 12-- The {text} facility option is undefined in the A to A protocol specification.

Condition Identifier: NFE\$UNKNOWN_FACILITY_OPTION Code: NF 12

Description: The facility option received by batch transfer services was not one permitted by the RHF A - A Protocol Specification.

User Action: Notify your site analyst.

--ERROR NF 13-- General state {text}, detail state {text} is undefined in the A to A protocol specification.

Condition Identifier: NFE\$UNKNOWN_TRANSFER_STATE Code: NF 13

Description: The RHF A - A Protocol provides for a parameter indicating the quality of the preceding file transfer to be exchanged after the data transfer stage. This message indicates receipt by batch transfer services of an improper value for this parameter.

User Action: Notify your site analyst.

--ERROR NF 14-- The {text} host type is undefined in the A to A specification.

Condition Identifier: NFE\$UNKNOWN_HOST_TYPE Code: NF 14

Description: The parameter received by batch transfer services describing host type contained a value not defined in the RHF A - A Protocol Specification.

User Action: Notify your site analyst.

--ERROR NF 15-- The {text} protocol identifier is undefined in the A to A specification.

Condition Identifier: NFE\$UNKNOWN_PROTOCOL Code: NF 15

Description: The parameter received by batch transfer services specifying the protocol version to be used contained a value not defined in the RHF A - A Protocol Specification.

User Action: Notify your site analyst.

--ERROR NF 16-- Illegal attempt to enter the transfer phase when a file transfer will not occur.

Condition Identifier: NFE\$ILLEGAL_NULL_TRANSFER Code: NF 16

Description: Batch Transfer Services attempted to enter the data transfer phase but the direction of transfer is unknown.

User Action: Notify your site analyst.

--ERROR NF 17-- Illegal event received on call to procedure {text}.

Condition Identifier: NFE\$ILLEGAL_EVENT Code: NF 17

Description: Batch Transfer Services received an unexpected network event.

User Action: Notify your site analyst.

--ERROR NF 18-- Procedure {text} abnormal condition {text}.

Condition Identifier: NFE\$BTS_INTERNAL_ERROR Code: NF 18

Description: A software error prevented Batch Transfer Services from continuing processing.

User Action: Notify your site analyst.

--ERROR NF 19-- The {text} command parameter length exceeds the maximum defined for the protocol.

Condition Identifier: NFE\$PARAM_DATA_SIZE_EXCEEDED Code: NF 19

Description: Batch Transfer Services discovered a parameter size field which exceeds the maximum size specified by the RHF A - A Protocol Specification.

User Action: Notify your site analyst.

--ERROR NF 20-- The address returned from title translation is of the wrong kind.

Condition Identifier: NFE\$INCOMPATIBLE_ADDRESS_KIND Code: NF 20

Description: The address type returned to Batch Transfer Services was not of the type expected.

User Action: Notify your site analyst.

--ERROR NF 21-- The negotiation to {text} was invalid for this transfer

Condition Identifier: NFE\$INVALID_PROTOCOL_NEGOT Code: NF 21

Description: The peer application attempted to negotiate the protocol up rather than down.

User Action: Notify your site analyst.

--ERROR NF 22-- The protocol identifier {text} is unknown to this application

Condition Identifier: NFE\$INVALID_PROTOCOL_VALUE Code: NF 22

Description: The protocol id value specified by the peer application is not valid.

User Action: Notify your site analyst.

--ERROR NF 23-- The character {text} is invalid for protocol parameter 26

Condition Identifier: NFE\$INVALID_P26_VALUE Code: NF 23

Description: An invalid value was specified/received for parameter 26 (job name) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 25-- The attempt to connect to location {text} failed, but is retryable

Condition Identifier: NFE\$RECOVERABLE_CONNECT Code: NF 25

Description: The applications's initial attempt at establishing a connection failed. Connection establishment may be retried.

User Action: None.

--ERROR NF 27-- The user PTFS job terminated prematurely

Condition Identifier: NFE\$USER_JOB_TERM_PREMATURELY Code: NF 27

Description: The PTF server job terminated before executing any of the specified user commands. This usually implies an error occurred in the user prolog.

User Action: If the PTF server job terminated while processing th user prolog, the output from the job will be printed. Therefore, the user should check for output on the remote system (if possible). Additionally, the user should check the prolog to ensure no errors exist.

--ERROR NF 28-- The user PTFS job timed out before obtaining connect

Condition Identifier: NFE\$USER_JOB_SWITCH_TIMEOUT Code: NF 28

Description: The PTF server job did not respond to the client job within the requested time interval. This interval is usually 10 minutes. This error implies that the server job was running on a saturated machine (i.e. a very busy system) or that commands executed on the remote system took a very long time. Some possible problems could include infinite, or very large, SCL loops, statements which cause long waits, or simply a procedure which compiles and links a large piece of software.

User Action: Recheck the directives specified for the remote system to ensure no large or infinite loops exist. If the directives look innocent, check with site administration on the loading of the remote system and the tuning priorities of the PTF server job class.

--ERROR NF 29-- An error occurred on the remote system, see job log for details

Condition Identifier: NFE\$REMOTE_SYSTEM_ERROR_SEE_JL Code: NF 29

Description: An error occurred on the remote system.

User Action: Check the local job log to determine the cause of the error.

--ERROR NF 30-- Login command is either missing or incorrect

Condition Identifier: NFE\$BAD_OR_MISSING_LOGIN_IN_JOB Code: NF 30

Description: The login command was not present or was incorrect.

User Action: Make sure the user/password combination exists on the remote system.

--ERROR NF 50-- Unspecific transfer.

Condition Identifier: NFE\$UNSPECIFIC_TRANSFER Code: NF 50

Description: The remote system encountered an error executing the remote directives.

User Action: Check the local job log to determine the cause of the error.

--ERROR NF 51-- Transfer rejected.

Condition Identifier: NFE\$TRANSFER_REJECTED_MESSAGE Code: NF 51

Description: The remote host could not complete the transfer successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 52-- Unacceptable transfer control attribute settings.

Condition Identifier: NFE\$UNACCEPTABLE_ATTRIBUTES Code: NF 52

Description: The remote host could not complete the transfer successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 53-- Unspecific file store.

Condition Identifier: NFE\$UNSPECIFIC_FILE_STORE Code: NF 53

Description: The remote host could not complete the transfer successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 54-- File not found or does not exist.

Condition Identifier: NFE\$FILE_NOT_FOUND Code: NF 54

Description: An operation on a remote file failed because the file did not exist or was not accessible.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 55-- No access to file quoted.

Condition Identifier: NFE\$NO_FILE_ACCESS Code: NF 55

Description: An operation on a remote file failed because the file was unaccessible.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 56-- Wrong file type.

Condition Identifier: NFE\$WRONG_FILE_TYPE Code: NF 56

Description: The remote system encountered an error trying to operate on a file.

User Action: Check the local job log to determine the cause of the error.

--ERROR NF 57-- File not available or off_line.

Condition Identifier: NFE\$FILE_UNAVAILABLE Code: NF 57

Description: An operation on a remote file failed because the file was not available at the current time.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 58-- Username unknown.

Condition Identifier: NFE\$INVALID_USER Code: NF 58

Description: The specified user name was incorrect or unavailable at the current time.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for validity of your username/password combination.

--ERROR NF 59-- Username and/or password not quoted or quoted incorrectly.

Condition Identifier: NFE\$INVALID_PASSWORD Code: NF 59

Description: The specified password was incorrect or unavailable at the current time.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for validity of your username/password combination.

--ERROR NF 60-- Account unknown.

Condition Identifier: NFE\$INVALID_ACCOUNT Code: NF 60

Description: The specified account was incorrect or unavailable at the current time.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for validity of your username/password combination.

--ERROR NF 61-- Account and/or password not quoted or quoted incorrectly.

Condition Identifier: NFE\$INVALID_ACCOUNT_PW Code: NF 61

Description: The specified account password was incorrect or unavailable at the current time.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for validity of your username/password combination.

--ERROR NF 62-- Account dollar limit exceeded.

Condition Identifier: NFE\$NO_MONEY Code: NF 62

Description: The specified user name has reached its resource limit.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for the resource limits of the user name.

--ERROR NF 63-- File size limit exceeded.

Condition Identifier: NFE\$FILE_TOO_LARGE Code: NF 63

Description: An attempt was made to transfer a file which was larger than the maximum for this file type.

User Action: Check the job log for messages from the remote system which further explain the remote status. This message is often encountered when transferring files to NOS or NOS/BE as indirect files (GET,SAVE) when they should be transferred as direct files (DEFINE,ATTACH).

--ERROR NF 64-- Output device unknown or unavailable.

Condition Identifier: NFE\$WRONG_DEVICE Code: NF 64

Description: The specified device is non_existent or not available.

User Action: Check the job log for messages from the remote system which further explain the remote status. Check with site administration for the status of the specified device.

--ERROR NF 66-- Terminated - refer to message text.

Condition Identifier: NFE\$TERMINATE_TRANSFER_MESSAGE Code: NF 66

Description: The remote host could not complete the transfer successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 67-- Accounting limit exceeded - discard file.

Condition Identifier: NFE\$ACCOUNTING_LIMIT_EXCEEDED Code: NF 67

Description: The remote host could not complete the specified directives because of a resource limit.

User Action: Check the job for messages from the remote system which further explain the remote status. Check with site administration for the limits on your account.

--ERROR NF 75-- Satisfactory and complete - No retry required.

Condition Identifier: NFE\$SATISFACTORY_AND_INCOMPLETE Code: NF 75

Description: The remote host could not complete the transfer successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 76-- Receiver problems - retry possible.

Condition Identifier: NFE\$RECEIVER_PROBLEM_RETRY Code: NF 76

Description: The receiver of a file in a file transfer could not complete successfully, however the file transfer may be tried again.

User Action: Retry the transfer. If it fails again, check the job log for messages which may further explain the situation.

--ERROR NF 77-- Receiver problems - no retry possible.

Condition Identifier: NFE\$RECEIVER_PROBLEM_NO_RETRY Code: NF 77

Description: The receiver of a file in a file transfer could not complete successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 78-- Sender problems - Retry possible.

Condition Identifier: NFE\$SENDER_PROBLEM_RETRY Code: NF 78

Description: The sender of a file in a file transfer could not complete successfully, however th file transfer may be tried again.

User Action: Retry the transfer. If it fails again, check the job log for messages which may further explain the situation.

--ERROR NF 79-- Sender problems - No retry possible.

Condition Identifier: NFE\$SENDER_PROBLEM_NO_RETRY Code: NF 79

Description: The sender of a file in a file transfer could not complete successfully.

User Action: Check the job log for messages from the remote system which further explain the remote status.

--ERROR NF 80-- Application time limit exceeded.

Condition Identifier: NFE\$APPLICATION_TIME_OUT Code: NF 80

Description: The server job did not complete the transfer within the time limit (10 minutes).

User Action: Recheck the directives specified for the remote system to ensure no large or infinite loops exist.
If the directives look innocent, check with site administration on the loading of the remote system and the tuning priorities of the PTF server job class.**--ERROR NF 81-- Irrecoverable protocol anomaly.**

Condition Identifier: NFE\$PROTOCOL_ANOMALY Code: NF 81

Description: The applications use of the A - A protocol is out of sequence or the protocol commands are invalid.

User Action: Notify your site_analyst.

--ERROR NF 82-- Invalid command parameter count {text} .

Condition Identifier: NFE\$INVALID_PARAM_COUNT Code: NF 82

Description: The parameter count value for a RHF A - A protocol command is invalid.

User Action: Notify your site analyst.

--ERROR NF 83-- Invalid protocol command {text} .

Condition Identifier: NFE\$INVALID_PROTOCOL_COMMAND Code: NF 83

Description: The protocol command is not one of the commands documented in the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 84-- Invalid protocol parameter 06, file size, {text} .

Condition Identifier: NFE\$INVALID_P06_VALUE Code: NF 84

Description: An invalid value was specified/received for parameter 6 (file length) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 85-- Invalid command parameter 12, max xfer block size {text} .

Condition Identifier: NFE\$INVALID_P12_VALUE Code: NF 85

Description: An invalid value was specified/received for parameter 12 (maximum block size) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 86-- Invalid command parameter 20, min time out {text} .

Condition Identifier: NFE\$INVALID_P20_VALUE Code: NF 86

Description: An invalid value was specified/received for parameter 20 (minimum timeout interval) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 87-- Invalid command parameter 21 mode {text} .

Condition Identifier: NFE\$INVALID_P21_VALUE Code: NF 87

Description: An invalid value was specified/received for parameter 21 (mode of access) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 88-- Invalid command parameter 21 specification {text} .

Condition Identifier: NFE\$INVALID_P21_SPEC Code: NF 88

Description: The mode option was not specified for parameter 21 (mode of access) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 89-- Invalid command parameter 22, host type {text} .

Condition Identifier: NFE\$INVALID_P22_VALUE Code: NF 89

Description: An invalid value was specified/received for parameter 22 (host type) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--WARNING NF 90-- Transfer in progress, user interrupt ignored.

Condition Identifier: NFE\$USER_INTERRUPT_IGNORED Code: NF 90

Description: The user attempted to pause-break the file transfer at an inappropriate time.

User Action: Wait for the file transfer to complete.

--ERROR NF 91-- protocol error {text}.

Condition Identifier: NFE\$DISLIKE_PARAMETER Code: NF 91

Description: An unacceptable facilities option was received and the RHF A - A protocol should not continue.

User Action: Notify your site analyst.

--ERROR NF 92-- transfer failed, begin retry.

Condition Identifier: NFE\$TRANSFER_FAILED_RECOVERING Code: NF 92

Description: The application detected an error during the file transfer that was recoverable.

User Action: None, recovery will be attempted.

--ERROR NF 93-- Invalid command parameter 17, disposition code {text} .

Condition Identifier: NFE\$INVALID_P17_VALUE Code: NF 93

Description: An invalid value was specified/received for parameter 17 (file disposition) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 94-- Invalid command parameter 18, acknowledgment window {text}

.

Condition Identifier: NFE\$INVALID_P18_VALUE Code: NF 94

Description: An invalid value was specified/received for parameter 18 (acknowledgment window) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 95-- Invalid command parameter 19, initial restart checkmark code {text} .

Condition Identifier: NFE\$INVALID_P19_VALUE Code: NF 95

Description: An invalid value was specified/received for parameter 19 (initial checkmark) of the RHF A - A protocol specification.

User Action: Notify your site analyst.

--ERROR NF 100-- PTF deactivated by operator request.

Condition Identifier: NFE\$PTF_DEACTIVATION_REQUESTED Code: NF 100

Description: PTF was deactivated by operator request.

User Action: None.

--ERROR NF 125-- The requested station/ device pair does not match the connected device.

Condition Identifier: NFE\$NON_MATCHING_DEVICE Code: NF 125

Description: Every instance of BTF executing is associated with a particular batch device. This message indicates a violation of that process.

User Action: Notify your site analyst.

--ERROR NF 150-- Asynchronous task not responding.

Condition Identifier: NFE\$TASK_NOT_RESPONDING Code: NF 150

Description: A request was placed in the request queue for the BTF task and no response was obtained in the allotted time.

User Action: Notify your site analyst.

--ERROR NF 151-- Asynchronous task timeout.

Condition Identifier: NFE\$ASYNC_TASK_TIMEOUT Code: NF 151

Description: A BTF child task was initiated with no entry in the asynchronous task directory.

User Action: Notify your site analyst.

--ERROR NF 153-- Bad message discarded.

Condition Identifier: NFE\$BAD_MESSAGE_DISCARDED Code: NF 153

Description: An improperly formatted message was being exchanged between BTF and SCF and was discarded by Common Task Communication.

User Action: Notify site analyst to handle possible system problem. Operations may be continued.

--ERROR NF 154-- Activity pending.

Condition Identifier: NFE\$ACTIVITY_PENDING Code: NF 154

Description: In trying to terminate a child task, Common Task Communication discovered unprocessed messages in the queues.

User Action: Notify your site analyst.

--ERROR NF 155-- Locked by another task.

Condition Identifier: NFE\$LOCKED_BY_ANOTHER_TASK Code: NF 155

Description: In trying to access the control structure for a task, Common Task Communication found the data area locked by another task.

User Action: Notify your site analyst.

--ERROR NF 156-- Module not initialized.

Condition Identifier: NFE\$MODULE_NOT_INITIALIZED Code: NF 156

Description: Call to Common Task Communication incorrect unless previous call had been made to initialize.

User Action: Notify your site analyst.

--ERROR NF 157-- Task not active.

Condition Identifier: NFE\$TASK_NOT_ACTIVE Code: NF 157

Description: Common Task Communication tried to send message to a task which apparently does not exist.

User Action: Notify your site analyst.

--ERROR NF 158-- Task not found in directory.

Condition Identifier: NFE\$TASK_NOT_FOUND Code: NF 158

Description: Common Task Communication is not able to find task in the task directory.

User Action: Notify your site analyst.

--ERROR NF 200-- Peer did not respond in allotted time.

Condition Identifier: NFE\$APPLICATION_TIMEOUT Code: NF 200

Description: BTF or PTF exceeded its timeout value while waiting for data from its peer during the data transfer phase of the file transfer.

User Action: Transfer can be retried. If condition repeats, notify site analyst.

--ERROR NF 201-- Message received from peer application contained protocol error.

Condition Identifier: NFE\$APPLICATION_PROTOCOL_ERROR Code: NF 201

Description: BTF or PTF received an incorrect command or an incorrect parameter within a command, given the current state of the transfer.

User Action: Transfer can be retried. If condition repeats, notify site analyst.

--ERROR NF 202-- Peer application unexpectedly closed connection.

Condition Identifier: NFE\$CONNECTION_CLOSED_BY_PEER Code: NF 202

Description: BTF or PTF received an indication that the peer application had broken the connection during the data transfer phase.

User Action: Transfer can be retried. If condition repeats, notify site analyst.

--ERROR NF 203-- Access method received no response from peer in allotted time.

Condition Identifier: NFE\$ACCESS_METHOD_TIMEOUT Code: NF 203

Description: BTF or PTF received an indication while waiting for data that the access method had timed out.

User Action: Transfer can be retried. If condition repeats, notify site analyst.

--ERROR NF 300-- Remote validation already defined for location {text}.

Condition Identifier: NFE\$REMOTE_VAL_DEFINED Code: NF 300

Description: A create remote validation request was made previously for the specified location.

User Action: Delete the current remote validation and enter a new one.

--ERROR NF 301-- No remote validation has been defined.

Condition Identifier: NFE\$REMOTE_VAL_EMPTY Code: NF 301

Description: A request was made to display or delete all validations and none were active.

User Action: None.

--ERROR NF 302-- Remote validation undefined for location {text}.

Condition Identifier: NFE\$REMOTE_VAL_UNDEFINED Code: NF 302

Description: A request was made to display or delete a validation for a particular location for which no validation was active.

User Action: If desired, create appropriate validation.

--ERROR NF 303-- Illegal device for remote validation display.

Condition Identifier: NFE\$REMOTE_VAL_DEVICE_ERROR Code: NF 303

Description: A request was made to display remote validation to a device other than an interactive terminal.

User Action: Correct the device type in the request.

--ERROR NF 305-- Attempted copy with both files remote.

Condition Identifier: NFE\$BOTH_FILES_REMOTE Code: NF 305

Description: Both input and output files of a copy_file command existed on remote mainframe(s). This is an illegal condition.

User Action: Correct the copy_file command and retry.

--ERROR NF 306-- Output for {text} may not be a remote file.

Condition Identifier: NFE\$DISPLAY_OUTPUT_REMOTE Code: NF 306

Description: A display command incorrectly specified the output file as a remote file.

User Action: Correct the output file parameter and retry the command.

--ERROR NF 307-- Remote file {text} not on mass storage device.

Condition Identifier: NFE\$REMOTE_FILE_NOT_MS Code: NF 307

Description: The user attempted to transfer a NOS/VE file which is not a mass storage device and is not mounted.

User Action: The user must ensure the device is mounted before the file transfer is initiated.

--ERROR NF 310-- {text} encountered EOI.

Condition Identifier: NFE\$ENCOUNTERED_EOI Code: NF 310

Description: Manage_remote_files reached end of information prior to encountering the specified ending characters.

User Action: Correct the text in the job directives and reenter.

--ERROR NF 311-- No server directives for {text}.

Condition Identifier: NFE\$DIRECTIVES_MISSING Code: NF 311

Description: The specified ending characters were reached without any intervening directives following a MANAGE_REMOTE_FILES command.

User Action: Reenter the MANAGE_REMOTE_FILES command and specify appropriate directives.

--ERROR NF 315-- Only one file transfer allowed per remote access.

Condition Identifier: NFE\$MULTIPLE_FILE_TRANSFERS Code: NF 315

Description: The PTF server job encountered more than one send_file and/or receive_file in the directive stream.

User Action: Correct the manage_remote_files directives and reenter the command.

--ERROR NF 350-- Operator Utility invalid intertask request.

Condition Identifier: NFE\$SOU_INVALID_INTERTASK_REQ Code: NF 350

Description: Operate_station received an invalid request from its child task helper.

User Action: Notify your site analyst.

--ERROR NF 351-- Operator Utility invalid intertask response.

Condition Identifier: NFE\$SOU_INVALID_INTERTASK_RESP Code: NF 351

Description: Operate_station received an invalid response from its child task helper.

User Action: Notify your site analyst.

--ERROR NF 352-- Operator Utility no response from asynchronous task.

Condition Identifier: NFE\$SOU_ASYNC_TASK_NO_RESPONSE Code: NF 352

Description: Operate_station exceeded its allotted time waiting for a response from its child task helper.

User Action: Notify your site analyst.

--ERROR NF 355-- User not validated to use {text}.

Condition Identifier: NFE\$SOU_INVALID_USER Code: NF 355

Description: User is not validated to use the specified service.

User Action: Use site procedures to obtain the necessary validation.

--ERROR NF 360-- Station Operator received no response from Control Facility, command timed out.

Condition Identifier: NFE\$SOU_SCFS_NO_RESPONSE Code: NF 360

Description: Operate_station received no response from SCFS within the allotted time.

User Action: Notify your site analyst.

--ERROR NF 361-- Format error in {text} message from Control Facility.

Condition Identifier: NF\$SOU_MESSAGE_FORMAT_ERROR Code: NF 361

Description: Operate_station received an improperly formatted message from SCFS.

User Action: Notify your site analyst.

--ERROR NF 362-- {text} message rejected, {text}.

Condition Identifier: NF\$SOU_MESSAGE_REJECT Code: NF 362

Description: Operate_station received a reject message from SCFS in response to a request.

User Action: Retry the request. If the problem persists, notify your site analyst.

--ERROR NF 363-- {text} request received from Control Facility.

Condition Identifier: NF\$SOU_UNEXPECTED_NETWORK_REQ Code: NF 363

Description: Operate_station received a non-data type network event.

User Action: Quit operate_station and enter the utility again. If the problem persists, notify your site analyst.

--ERROR NF 364-- {text} command rejected, {text}.

Condition Identifier: NF\$SOU_COMMAND_REJECT Code: NF 364

Description: The subcommand entered via the Operate_Station utility was rejected by the I/O station control facility.

User Action: Check the command previously entered. If the command contains errors, reenter the command correctly.

--ERROR NF 365-- Invalid value for LOCATION parameter. Integer (0..65535) or list of string (1..255) allowed.

Condition Identifier: NF\$INVALID_VALUE_FOR_LOCATION Code: NF 365

Description: The location parameter for the position_file command was specified incorrectly.

User Action: Refer to the documentation on the position_file command and reenter the command, specifying the location parameter correctly.

--ERROR NF 367-- {text} command rejected, error on attribute {text}.

Condition Identifier: NF\$ATTRIBUTE_ERROR_ON_COMMAND Code: NF 367

Description: The attribute change requested by the station operator for a specific device was rejected by the DI.

User Action: Check that the change requested is allowed for the device specified, as it is currently configured. If it should be allowed, notify your site analyst.

--ERROR NF 368-- Cannot locate I/O station {text}.

Condition Identifier: NF\$STATION_TITLE_NOT_ACTIVE Code: NF 368

Description: The specified I/O station name could not be found on the network in the allotted time (approximately 2 minutes).

User Action: The I/O station that operation is requested for may not be configured yet, or it may be in the process of being configured. It may also be that the control facility controlling the specified I/O station didn't respond fast enough. If the error persists after repeated attempts and the station specified should exist, notify your site analyst.

--ERROR NF 400-- Control facility title, {text}, is already registered.

Condition Identifier: NF\$SCF_TITLE_ALREADY_REGISTERED Code: NF 400

Description: SCFS cannot be activated since an SCFS for the control facility is already active.

User Action: None.

--ERROR NF 401-- Attribute request is not allowed by the device.

Condition Identifier: NFE\$INVALID_CHG_REQU_BY_OPER Code: NF 401

Description: The change batch device attributes message sent by SCFS/VE was rejected by the DI.

User Action: Check that the change requested is allowed for the device specified, as it is currently configured. If it should be allowed, notify your site analyst.

--ERROR NF 450-- Pointer outside the bounds of the message sequence.

Condition Identifier: NFE\$OUTSIDE_BOUNDS_OF_SEQUENCE Code: NF 450

Description: A read/write beyond the allocated message space was attempted while parsing/creating a SCFS protocol message.

User Action: Notify your site analyst.

--ERROR NF 451-- Incorrect or missing parameter value {text} for {text} protocol message.

Condition Identifier: NFE\$INVALID_PARAMETER_VALUE Code: NF 451

Description: An invalid value for a parameter was received on a SCFS protocol message.

User Action: Notify your site analyst.

--ERROR NF 452-- Descriptor value {text} invalid for application.

Condition Identifier: NFE\$INVALID_DESCRIPTOR_VALUE Code: NF 452

Description: An invalid input/output descriptor value was specified as input to the SCFS protocol message generation procedures.

User Action: Check the queue descriptor being passed to the procedures. Correct the appropriate field(s), as indicated by the error message.

--ERROR NF 453-- Queue file descriptors must be of the same type.

Condition Identifier: NFE\$CONFLICTING_DESCRIPTOR Code: NF 453

Description: Queue file descriptors are not of the same type.

User Action: Correct the procedure call to specify either input descriptors or output descriptors.

--ERROR NF 500-- The explicit text directive {text} is not a legal directive.

Condition Identifier: NFE\$INVALID_EXPLICIT_TEXT Code: NF 500

Description: An explicit text directive was specified to QTFS which was not legal or was invalid.

User Action: NOS/VE QTFS accepts the PRINT_FILE directive for output and the SUBMIT_JOB directive for jobs. Make sure the directive follows the correct syntax for the command.

--ERROR NF 501-- The data declaration US may not be specified for a job.

Condition Identifier: NFE\$DATA_TYPE_US_NOT_FOR_JOB Code: NF 501

Description: The data declaration for the job file transfer was specified as US.

User Action: The data declaration for a job file transfer may not be US, restructure the job to use a different data declaration.

--ERROR NF 502-- The data mode {text} is invalid.

Condition Identifier: NFE\$INVALID_DATA_MODE_PARAMETER Code: NF 502

Description: The data mode parameter value specified on a PRINT_FILE command was incorrect.

User Action: Refer to documentation for the PRINT_FILE command and correct the data mode value.

--ERROR NF 503-- The vertical print density parameter {text} is invalid.

Condition Identifier: NFE\$INVALID_VPD_PARAMETER Code: NF 503

Description: The vertical_print_density parameter value specified on a PRINT_FILE command was incorrect.

User Action: Refer to documentation for the PRINT_FILE command and correct the vertical_print_density value.

--ERROR NF 504-- The implicit text directive {text} is invalid.

Condition Identifier: NFE\$INVALID_IMPLICIT_TEXT_VALUE Code: NF 504

Description: The implicit text for the queue file transfer is in the incorrect format.

User Action: Notify your site analyst.

--ERROR NF 505-- The copies parameter specified is out of range.

Condition Identifier: NFE\$COPIES_OUT_OF_RANGE Code: NF 505

Description: The copies parameter value specified on a PRINT_FILE command was incorrect.

User Action: Refer to documentation for the PRINT_FILE command and correct the copies parameter.

--ERROR NF 506-- The job execution ring specified is out of range.

Condition Identifier: NFE\$JOB_EXEC_RING_OUT_OF_RANGE Code: NF 506

Description: The JOB_EXECUTION_RING parameter value specified on a SUBMIT_JOB command was out of range.

User Action: Refer to documentation for the PRINT_FILE command and correct the JOB_EXECUTION_RING parameter.

--ERROR NF 510-- A communication error occurred between QTFC and QTFI.

Condition Identifier: NFE\$QTFC_QTFI_COMMUNICATION_ERR Code: NF 510

Description: The QTF client system task had an error in communicating with the child task that does the actual file transfer.

User Action: Restart the QTF client system task and notify your site analyst.

--ERROR NF 511-- Job loopback transfers not permitted.

Condition Identifier: NFE\$QTF_NO_LOOPBACK_JOBS Code: NF 511

Description: QTF is receiving an input queue file that originated on this system. This could produce an infinite loop situation.

User Action: If a batch job is going to execute for a family that is defined on the current system that you are executing on, do not submit a job with a JOB_DESTINATION_USAGE of QTF.

--ERROR NF 512-- NOS/VE QTF does not support the printing of NON-NOS/VE files to the WAIT QUEUE.

Condition Identifier: NFE\$PRIF_TO_WAIT_QUEUE_NOT_SUP Code: NF 512

Description: QTF is receiving an output queue file that originated on a non-NOS/VE system by a user printing a file designated for a NOS/VE wait queue.

User Action: None.

--INFORMATIVE NF 513-- NOS/VE QTF SERVER has changed the OUTPUT_DESTINATION_USAGE for the system job/file name of + P1 from the JOB_ATTRIBUTE_DEFAULT of QTF to PUBLIC.

Condition Identifier: NFE\$QTFS_CHG_JAD_QTF_TO_PUBLIC Code: NF 513

Description: QTF is receiving a queue file destined for this system and the site analyst has define the system's job attribute default for OUTPUT_DESTINATION_USAGE to be QTF.

User Action: None.

--FATAL NF 600-- Caller not validated to {text}.

Condition Identifier: NFE\$SF_CALLER_NOT_PRIVILEGED Code: NF 600

Description: To use the MANAGE_STORE_FORWARD_NETWORK utility, you must be validated for the capability of NETWORK_OPERATION or NETWORK_APPLICATION_MANAGEMENT.

User Action: Ask the site administrator to validate you for the NETWORK_OPERATION or NETWORK_APPLICATION_MANAGEMENT capability.

--INFORMATIVE NF 601-- Combine the multiple definitions of DEFINE_APPLICATION_NAME_SWITCH, that have the destination_group_qualifier value of {text}, next_hop_application value of {text} and the application_qualifiers are a subset of one or the other into one definition.

Condition Identifier: NFESF_COMBINE_DUP_APPL_DEF Code: NF 601

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_APPLICATION_NAME_SWITCH directives that contain a subset of identical information.

User Action: Combine all applicable directives into one directive.

--INFORMATIVE NF 602-- Combine the multiple definitions of DEFINE_SOURCE_NAME_SWITCH, that have the name value of {text}, next_hop_name value of {text}, destination_group_qualifier value of {text}, and the application_qualifiers are a subset of one another into one definition.

Condition Identifier: NFESF_COMBINE_DUP_SOURCE_DEF Code: NF 602

Description: @ The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_SOURCE_NAME_SWITCH directives that contain a subset of identical information.

User Action: Combine all applicable directives into one directive.

--INFORMATIVE NF 603-- Combine the multiple definitions of DEFINE_DESTINATION_NAME_SWITCH, that have the name value of {text}, next_hop_name value of {text} and the application_qualifiers are a subset of one another into one definition.

Condition Identifier: NFESF_COMBINE_DUP_TARGET_DEF Code: NF 603

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_DESTINATION_NAME_SWITCH directives that contain a subset of identical information.

User Action: Combine all applicable directives into one directive.

--FATAL NF 604-- The DESTINATION_GROUP_QUALIFIER {text} has not been defined.

Condition Identifier: NFESF_DEST_GROUP_NOT_FOUND Code: NF 604

Description: The destination group qualifier has not been defined.

User Action: Define the destination group by using the MANAGE_STORE_FORWARD_NETWORK directive DEFINE_DESTINATION_GROUP.

--FATAL NF 605-- The multiple definitions of DEFINE_APPLICATION_NAME_SWITCH, that have the destination_group_qualifier value of {text} and overlapping application_qualifiers have different next_hop_applications.

Condition Identifier: NFESF_DIF_NHA_SAME_AQ_AND_DGQ Code: NF 605

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_APPLICATION_NAME_SWITCH directives that contain identical values for the parameters APPLICATION_QUALIFIER and DESTINATION_GROUP_QUALIFIER but different values for the parameter NEXT_HOP_APPLICATION.

User Action: Change the NEXT_HOP_APPLICATION to the correct value or change the APPLICATION_QUALIFIER or DESTINATION_GROUP_QUALIFIER parameters to different values.

--FATAL NF 606-- The multiple definitions of DEFINE_SOURCE_NAME_SWITCH, that have the name value of {text}, destination_group_qualifier value of {text} and overlapping application_qualifiers have different next_hop_names.

Condition Identifier: NFESF_DIF_NHN_SAME_N_AQ_DGQ Code: NF 606

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_SOURCE_NAME_SWITCH directives that contain identical values for the parameters NAME, APPLICATION_QUALIFIER, and DESTINATION_GROUP_QUALIFIER but different values for the parameter NEXT_HOP_NAME.

User Action: Change the NEXT_HOP_NAME to the correct value or change the NAME or APPLICATION_QUALIFIER or DESTINATION_GROUP_QUALIFIER parameters to different values.

--FATAL NF 607-- The multiple definitions of DEFINE_DESTINATION_NAME_SWITCH, that have the name value of {text} and overlapping application_ qualifiers have different next_hop_names.

Condition Identifier: NF\$SF_DIF_NHN_SAME_N_AND_AQ Code: NF 607

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_DESTINATION_NAME_SWITCH directives that contain identical values for the parameters NAME and APPLICATION_QUALIFIER but different values for the parameter NEXT_HOP_NAME.

User Action: Change the NEXT_HOP_NAME to the correct value or change the NAME or APPLICATION_QUALIFIER parameters to different values.

--FATAL NF 608-- MANAGE_STORE_FORWARD_NETWORK has encountered directive errors, see file: {text} for additional error information.

Condition Identifier: NF\$SF_DIRECTIVE_ERRORS Code: NF 608

Description: The MANAGE_STORE_FORWARD_NETWORK utility has encountered fatal directive errors that cannot be processed.

User Action: Look at the output file for the MANAGE_STORE_FORWARD_NETWORK utility for additional information about the directive errors, and then correct the errors.

--FATAL NF 609-- The DEFINE_APPLICATION_NAME_SWITCH, that have the destination_group_qualifier value of {text} and next_hop_application value of {text} has multiple definitions with overlapping application_ qualifiers.

Condition Identifier: NF\$SF_DUPLICATE_APPL_DEF Code: NF 609

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_APPLICATION_NAME_SWITCH directives for the same NEXT_HOP_APPLICATION and DESTINATION_GROUP_QUALIFIER values but different APPLICATION_QUALIFIER values.

User Action: Delete the invalid directive.

--FATAL NF 610-- The DEFINE_DESTINATION_GROUP, that has the group_ name value of {text} has multiple definitions.

Condition Identifier: NF\$SF_DUPLICATE_GROUP_NAMES Code: NF 610

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_DESTINATION_GROUP directives for the same GROUP_NAME value.

User Action: Delete the invalid directive.

--FATAL NF 611-- The DEFINE_SOURCE_NAME_SWITCH, that have the name value of {text} and next_hop_name value of {text} has multiple definitions with overlapping application_ qualifiers and destination_group_ qualifiers.

Condition Identifier: NF\$SF_DUPLICATE_SOURCE_DEF Code: NF 611

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_SOURCE_NAME_SWITCH directives for the same NAME and NEXT_HOP_NAME values but different APPLICATION_QUALIFIER and DESTINATION_GROUP_QUALIFIER values.

User Action: Delete the invalid directive.

--FATAL NF 612-- The DEFINE_DESTINATION_NAME_SWITCH, that have the name value of {text} and next_hop_name value of {text} has multiple definitions with overlapping application_ qualifiers.

Condition Identifier: NF\$SF_DUPLICATE_TARGET_DEF Code: NF 612

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_DESTINATION_NAME_SWITCH directives for the same NAME and NEXT_HOP_NAME values but different APPLICATION_QUALIFIER . values.

User Action: Delete the invalid directive.

--INFORMATIVE NF 613-- The DEFINE_DESTINATION_GROUP directive for the group_name {text} was not used as a destination_group_qualifier in any other directives.

Condition Identifier: NFE\$SF_GROUP_NAME_NOT_USED Code: NF 613

Description: The MANAGE_STORE_FORWARD_NETWORK utility DEFINE_DESTINATION_GROUP directive is not used in any other directive.

User Action: Delete the unused directive.

--INFORMATIVE NF 614-- The DEFINE_DESTINATION_GROUP, that has the group_name {text} has the identical destination names as group_name {text}.

Condition Identifier: NFE\$SF_IDENTICAL_DEST_LIST Code: NF 614

Description: The MANAGE_STORE_FORWARD_NETWORK utility has found multiple DEFINE_DESTINATION_GROUP directives with different GROUP_NAME values but identical destination names.

User Action: Correct the directives to use only one GROUP_NAME and delete the unused directive.

--FATAL NF 615-- An invalid pointer type has been encountered in the procedure: {text}

Condition Identifier: NFE\$SF_INTERNAL_ERROR_BAD_PTR Code: NF 615

Description: An internal error has occurred in the MANAGE_STORE_FORWARD_NETWORK utility.

User Action: Reinstall the MANAGE_STORE_FORWARD_NETWORK directive file.

--CATASTROPHIC NF 616-- An invalid pointer type has been encountered in the procedure: {text} please re-install the SYSTEM's Store/Forward Network File.

Condition Identifier: NFE\$SF_INTERNAL_ERROR_BAD_RPTR Code: NF 616

Description: An internal error has occurred in the MANAGE_STORE_FORWARD_NETWORK utility.

User Action: Reinstall the MANAGE_STORE_FORWARD_NETWORK directive file.

--ERROR NF 617-- The name value {text} must be no longer than 31 characters in length for a {text}.

Condition Identifier: NFE\$SF_NAME_TOO_LONG Code: NF 617

Description: A name value supplied for a MANAGE_STORE_FORWARD_NETWORK directive parameter is too long.

User Action: Shorten the name to a value less than or equal to 31 characters.

--ERROR NF 618-- The name value {text} must be at least 3 characters in length for a {text}.

Condition Identifier: NFE\$SF_NAME_TOO_SHORT Code: NF 618

Description: A name value supplied for a MANAGE_STORE_FORWARD_NETWORK directive parameter is too short.

User Action: Lengthen the name to a value greater than or equal to 3 characters.

--FATAL NF 619-- The System's Store/Forward Network File does not exist, please install it.

Condition Identifier: NFE\$SF_NO_STORE_FORWARD_NETWORK Code: NF 619

Description: The MANAGE_STORE_FORWARD_NETWORK directive file has not been installed.

User Action: Install the MANAGE_STORE_FORWARD_NETWORK directive file.

--CATASTROPHIC NF 620-- Errors encountered on reading the System's Store/Forward Network File.

Condition Identifier: NFE\$SF_READ_NETWORK_FILE_ERROR Code: NF 620

Description: The installed version of the MANAGE_STORE_FORWARD_NETWORK directive file has been corrupted.

User Action: Reinstall the MANAGE_STORE_FORWARD_NETWORK directive file.

--ERROR NF 621-- The string value {text} must be no longer than 31 characters in length for a {text}.

Condition Identifier: NFE\$SF_STRING_TOO_LONG Code: NF 621

Description: A string value supplied for a MANAGE_STORE_FORWARD_NETWORK directive parameter is too long.

User Action: Shorten the string to a value less than or equal to 31 characters.

--ERROR NF 622-- The string value {text} must be at least 3 characters in length for a {text}.

Condition Identifier: NFE\$SF_STRING_TOO_SHORT Code: NF 622

Description: A string value supplied for a MANAGE_STORE_FORWARD_NETWORK directive parameter is too short.

User Action: Lengthen the string to a value greater than or equal to 3 characters.

--CATASTROPHIC NF 623-- Installation of System's Store/Forward Network File failed due to errors encountered on writing the Store/Forward Network File.

Condition Identifier: NFE\$SF_WRITE_NETWORK_FILE_ERROR Code: NF 623

Description: The MANAGE_STORE_FORWARD_NETWORK directive file cannot be installed due to system errors.

User Action: Try to reinstall the MANAGE_STORE_FORWARD_NETWORK directive file.

--ERROR OC 0-- Module {text} not found.

Condition Identifier: OCE\$E_MODULE_NOT_FOUND Code: OC 0

Description: The specified module is not in the current module list.

User Action: Add the module to the module list and then reenter the CHANGE_MODULE_ATTRIBUTES command.

--ERROR OC 1-- Internal OLG segment overflow.

Condition Identifier: OCE\$E_INTERNAL_OLG_SEG_OVERFLOW Code: OC 1

Description: Internal object library generator (OLG) error.

User Action: Inform a site analyst of the error with appropriate documentation of the error.

--ERROR OC 2-- Premature end of file encountered on file {file}.

Condition Identifier: OCE\$E_PREMATURE_EOF_ON_FILE Code: OC 2

Description: Object library format error. The object library generator read the end of the file before it read the final module

User Action: Regenerate the object library using the original modules.

--ERROR OC 3-- Module {text} already on current library.

Condition Identifier: OCE\$E_MODULE_ALREADY_ON_LIBRARY Code: OC 3

Description: The specified module name cannot be used because a module having that name already exists in the module list.

User Action: Reenter the command, specifying another name for the module.

--ERROR OC 4-- Too many components specified.

Condition Identifier: OCE\$E_TOO_MANY_COMPONENTS Code: OC 4

Description: Module format error. The load module had more than 0FFFF(16) components.

User Action: Generate a load module with fewer components if possible.

--ERROR OC 5-- Entry point {text} already exists.

Condition Identifier: OCE\$E_XDCL_ALREADY_EXISTS Code: OC 5

Description: The specified entry point cannot be created because an entry point having that name already exists in the module. Entry points having the same name and the retain or gate attribute exist in more than one component module.

User Action: Either remove the retain or gate attribute from the entry point in one of the component modules or change the entry point name.

--ERROR OC 6-- Entry point {text} not found.

Condition Identifier: OCE\$E_XDCL_DOESNT_EXIST Code: OC 6

Description: The entry point cannot be retained because it does not exist in the component modules.

User Action: Reenter the CREATE_MODULE command, specifying only entry points that exist in the component modules.

--ERROR OC 7-- Module {text} specified in both MODULE and LOCATION parameters.

Condition Identifier: OCE\$E_AFTER_IN_REORDER_LIST Code: OC 7

Description: You specified the same module on the LOCATION parameter and in the module list on the MODULE parameter.

User Action: Either specify another module to indicate the location where the reordered modules are to be inserted or remove the module from the list of reordered modules.

--ERROR OC 8-- Multiple identification records found in module {text}.

Condition Identifier: OCE\$E_MULTIPLE_IDENT_REC Code: OC 8

Description: Module format error. The module has more than one identification record pair.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 9-- Record number {text} in module {text} is an unknown record kind.

Condition Identifier: OCE\$E_INVALID_OBJECT_REC_KIND Code: OC 9

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 10-- Identification record expected on load module {text}.

Condition Identifier: OCE\$E_NO_IDENT_REC_ON_LOAD_MOD Code: OC 10

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 11-- Code section {text} not found in module {text}.

Condition Identifier: OCE\$E_CODE_SECTION_NOT_FOUND Code: OC 11

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 12-- Object file {file} is empty.

Condition Identifier: OCE\$E_EMPTY_OBJECT_FILE Code: OC 12

Description: The specified file is empty.

User Action: Reenter the command specifying the correct file.

--ERROR OC 13-- Unknown section ordinal encountered in module {text}.

Condition Identifier: OCE\$E_INVALID_SECTION_ORDINAL Code: OC 13

Description: Module format error. A valid section ordinal must be within the range 0 through 65,535.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 14-- Duplicate section definition encountered in module {text}.

Condition Identifier: OCE\$E_DUPLICATE_SECTION_DEFN Code: OC 14

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 15-- Missing section definition in module {text}.

Condition Identifier: OCE\$E_SECTION_NOT_YET_DEFINED Code: OC 15

Description: Module format error. A section definition record pair is missing on the module.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 16-- Referencing outside section in module {text}.

Condition Identifier: OCE\$E_REFERENCE_OUTSIDE_SECTION Code: OC 16

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 17-- Section definition expected in module {text}.

Condition Identifier: OCE\$E_SECTION_DEFN_EXPECTED Code: OC 17

Description: Module format error. For more information, refer to the object module format appendix in the NOS/VE Object Code Management manual.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 18-- Missing code section definition in module {text}.

Condition Identifier: OCE\$E_MISSING_CODE_SECTION_DEFN Code: OC 18

Description: Module format error. A section definition record pair is missing on the module.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 19-- Relocation record expected in module {text}.

Condition Identifier: OCE\$E_RELOCATION_REC_EXPECTED Code: OC 19

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 20-- Too many relocation items encountered in module {text}.

Condition Identifier: OCE\$E_TOO_MANY_RELOCATION_ITEMS Code: OC 20

Description: Module format error. The module contains more than 0FFFF(16) relocation items.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 21-- Too many template items encountered in module {text}.

Condition Identifier: OCE\$E_TOO_MANY_TEMPLATE_ITEMS Code: OC 21

Description: Module format error. The module contains more than 65,535 template items.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 22-- Too many modules on library.

Condition Identifier: OCE\$E_TOO_MANY_MODULES_ON_LIB Code: OC 22

Description: The module list contained more modules than the maximum allowed in the module directory of an object library. The maximum modules allowed is 65,535.

User Action: Reduce the number of modules.

--ERROR OC 23-- Too many entry points encountered.

Condition Identifier: OCE\$E_TOO_MANY_ENTRY_POINTS Code: OC 23

Description: The modules contained more entry points than the maximum allowed in the entry point directory of an object library. The maximum entry points allowed in 16,777,216.

User Action: Remove one or more modules from the module list and reenter the GENERATE_LIBRARY command.

--ERROR OC 24-- Too many section definitions encountered.

Condition Identifier: OCE\$E_TOO_MANY_SECTION_DEFNS Code: OC 24

Description: Module format error. The module contains more than 65,535 section definition record pairs.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 25-- File {file} specified is not a library.

Condition Identifier: OCE\$E_OBJECT_FILE_MUST_BE_LIB Code: OC 25

Description: A file specified on the LIBRARY parameter is not an object library. Each file specified must be an object library so that it contains an entry point directory for the command to search.

User Action: Reenter the command, specifying only object libraries on the LIBRARY parameter.

--ERROR OC 26-- Conflicting on-off toggles specified.

Condition Identifier: OCE\$E_ON_OFF_TOGGLES_OVERLAP Code: OC 26

Description: You specified conflicting display options. If you specify the keyword value ALL or NONE, do not specify other keyword values.

User Action: Correct and reenter the command.

--ERROR OC 27-- No modules on current library.

Condition Identifier: OCE\$E_NO_MODULES_ON_CURRENT_LIB Code: OC 27

Description: The command could not be processed because the current module list contains no modules. Note that the module list is cleared after processing of a GENERATE_LIBRARY command.

User Action: Add modules to the module list as needed.

--ERROR OC 28-- File name {file} can not be part of a subrange {file} .. {text}.

Condition Identifier: OCE\$E_FILE_NAME_CANNOT_BE_RANGE Code: OC 28

Description: An ellipsis (..) follows a file name in the component list.

User Action: Reenter the command with a correct component list. Each item in the list begins with a file name. If you do not want to use all modules on the file, follow the file name with a comma and a module list. The module list can include one or more subranges of modules. Enclose each item in the component list and the entire list in parentheses.

--ERROR OC 29-- Module {text} not on current library.

Condition Identifier: OCE\$E_MODULE_NOT_ON_LIBRARY Code: OC 29

Description: The command could not be processed because the specified module is not in the module list.

User Action: Either add the module to the module list or specify a module on the command that is in the module list.

--ERROR OC 30-- Identification record expected on object file {file}.

Condition Identifier: OCE\$E_NO_IDENT_REC_ON_OBJ_FILE Code: OC 30

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 31-- Empty or missing file {file}.

Condition Identifier: OCE\$E_MISSING_OR_EMPTY_FILE Code: OC 31

Description: Either the specified file is empty or it cannot be found.

User Action: Reenter the command specifying the correct file.

--ERROR OC 32-- File {file} does not contain object code.

Condition Identifier: OCE\$E_NON_OBJECT_FILE Code: OC 32

Description: The specified file is not an object file nor an object library.

User Action: Reenter the command, specifying an object file or object library.

--ERROR OC 33-- Premature end of file encountered in module {text}.

Condition Identifier: OCE\$E_PREMATURE_EOF_IN_MODULE Code: OC 33

Description: Module format error. The object library generator read the end of the file before it read a transfer symbol record for the module.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 34-- Non object module {text} cannot be placed on an object file.

Condition Identifier: OCE\$E_NON_OBJECT_MODULE_FOUND Code: OC 34

Description: Module format error. Data on the object file does not conform to the object module format conventions.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 35-- Too many libraries encountered.

Condition Identifier: OCE\$E_TOO_MANY_LIBRARIES Code: OC 35

Description: Module format error. The module specified more than 65,535 object libraries to be added to the program library list.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 36-- Relocation container goes beyond record area in module {text}.

Condition Identifier: OCE\$E_CONTAINER_OUTSIDE_RECORD Code: OC 36

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 37-- Offset for instruction opcode is less than zero in module {text}.

Condition Identifier: OCE\$E_OPCODE_NOT_WITHIN_RECORD Code: OC 37

Description: Module format error. The module contains a negative offset value.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 38-- Unknown relocation container kind encountered in module {text}.

Condition Identifier: OCE\$E_INVALID_CONTAINER_KIND Code: OC 38

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 39-- Sign bit set in relocation container for signed address type in module {text}.

Condition Identifier: OCE\$E_SIGN_BIT_SET_IN_CONTAINER Code: OC 39

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 40-- Unknown relocation container address type in module {text}.

Condition Identifier: OCE\$E_INVALID_CONTAINER_ADR_TYP Code: OC 40

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 41-- Unknown binding template address kind encountered in module {text}.

Condition Identifier: OCE\$E_INVALID_TEMPLATE_ADR_KIND Code: OC 41

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 42-- Relocating section is not code or binding in module {text}.

Condition Identifier: OCE\$E_RELOCATING_SEC_WRONG_KIND Code: OC 42

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 43-- Starting procedure {text} not located in a code section.

Condition Identifier: OCE\$E_STARTING_PROC_NOT_IN_CODE Code: OC 43

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 44-- GENERATE is aborting - no recovery is possible.

Condition Identifier: OCE\$E_GENERATE_IS_ABORTING Code: OC 44

Description: Internal object library generator error.

User Action: Provide a site analyst with documentation of the error.

--ERROR OC 45-- Offset arithmetic in bind section of module {text}.

Condition Identifier: OCE\$E_OFFSET_ARITHMETIC_IN_BS Code: OC 45

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 46-- Binding offset has no binding section template with matching offset in module {text}.

Condition Identifier: OCE\$E_BAD_BINDING_SEC_OFFSET Code: OC 46

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 47-- Unknown binding template address kind encountered.

Condition Identifier: OCE\$E_INVALID_ADDRESS_KIND Code: OC 47

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 48-- No modules on library {file}.

Condition Identifier: OCE\$E_NO_MODULES_ON_LIBRARY Code: OC 48

Description: The command could not display the modules in the file because the file contains no modules.

User Action: Check that the correct file name is specified on the command.

--ERROR OC 49-- Incompatible library module kind for module {text}.

Condition Identifier: OCE\$E_INVALID_MODULE_KIND Code: OC 49

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 50-- Too many modules on file {file}.

Condition Identifier: OCE\$E_TOO_MANY_MODULES_ON_FILE Code: OC 50

Description: The module list contained more modules than the maximum allowed in the module directory of an object library. The maximum modules allowed is 65,535.

User Action: Remove one or more modules from the module list and reenter the GENERATE_LIBRARY command.

--ERROR OC 51-- Map option parameter NONE used with other specifications.

Condition Identifier: OCE\$E_CONFLICTING_MAP_OPTIONS Code: OC 51

Description: You specified the keyword value NONE with other keyword values on the LOAD_MAP_OPTIONS parameter.

User Action: Reenter the command, either omitting the NONE keyword value or specifying it alone.

--ERROR OC 52-- File {file} previously opened as an object file.

Condition Identifier: OCE\$E_FILE_OPENED_AS_OBJECT Code: OC 52

Description: The object library generator cannot use the specified file as the output file because you previously specified the file as an input file containing modules for the new library.

User Action: Reenter the command, specifying another file on the OUTPUT parameter.

--ERROR OC 53-- End of file encountered on generated library.

Condition Identifier: OCE\$E_EOF_ON_GENERATED_FILE Code: OC 53

Description: While writing the generated library, the object library generator encountered the end of the file. This could occur if the new object library is longer than your maximum file size validation or if you do not have append access to the file.

User Action: Either decrease the size of the object library to be generated or enable the object library generator to write a longer file by obtaining append access or a larger file size validation.

--ERROR OC 54-- Module {text} being bound is type nonbindable.

Condition Identifier: OCE\$E_MODULE_NOT_BINDABLE Code: OC 54

Description: Module format error. The identification record of the module indicates it has the nonbindable attribute.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 55-- Duplicate entry point {text} found on libraries.

Condition Identifier: OCE\$E_DUPLICATE_GLOBAL_EPT Code: OC 55

Description: The entry point directories of the libraries contain more than one entry naming the same entry point.

User Action: Eliminate the duplication.

--ERROR OC 56-- External linkage record not found for binding section in module {text}.

Condition Identifier: OCE\$E_BND_SEC_EXT_NOT_FOUND Code: OC 56

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 57-- Incompatible object library version {text} on library {file}.

Condition Identifier: OCE\$E_INVALID_LIBRARY_VERSION Code: OC 57

Description: The object library version as recorded in the library header is incompatible with the current version of the object library generator.

User Action: If possible, regenerate the object library using the original modules.

--ERROR OC 58-- Incompatible object text version {text} in module {text}.

Condition Identifier: OCE\$E_INVALID_OBJ_TEXT_VERSION Code: OC 58

Description: The object text version as recorded in the module header is incompatible with the current version accepted by the loader and object library generator.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 59-- SCL procedure number {text} on file {file} is incorrect or the file attribute(s) are incompatible.

Condition Identifier: OCE\$E_INVALID_SCL_PROC Code: OC 59

Description: When attempting to generate procedure modules from the text on the specified file, the object library generator determined that the form of the specified procedure is incorrect.

User Action: List the procedure text and correct its errors.

--ERROR OC 60-- Module {text} not found - module ignored.

Condition Identifier: OCE\$W_MODULE_NOT_FOUND Code: OC 60

Description: The specified module was not found and therefore, was not processed.

User Action: Ensure that the module is specified correctly.

--ERROR OC 61-- Module {text} not found - Module subrange {text} .. {text} ignored.

Condition Identifier: OCE\$W_SUBRANGE_MODULE_NOT_FOUND Code: OC 61

Description: A module specifying the lower bound of a subrange is not in the file. Therefore, the entire module subrange was not processed.

User Action: Reenter the command, specifying the correct module subrange. To list the modules in the file, enter a DISPLAY_OBJECT_LIBRARY command.

--ERROR OC 62-- Duplicate module {text} specified - second occurrence ignored.

Condition Identifier: OCE\$W_SAME_MODULE_QUOTED_TWICE Code: OC 62

Description: You specified the same module twice on the command. The first occurrence was processed and the second ignored.

User Action: None.

--ERROR OC 63-- Module {text} already on current library - module ignored.

Condition Identifier: OCE\$W_MODULE_ALREADY_ON_LIBRARY Code: OC 63

Description: The specified module is already in the current module list. Therefore, the module cannot be added.

User Action: If needed, you can replace the existing module with the new module by entering a REPLACE_MODULES command.

--ERROR OC 64-- Module {text} not on current library - module ignored.

Condition Identifier: OCE\$W_MODULE_NOT_ON_LIBRARY Code: OC 64

Description: The specified module was not deleted because it is not in the current module list.

User Action: None.

--ERROR OC 65-- Module {text} not on current library - module subrange {text} .. {text} ignored.

Condition Identifier: OCE\$W_SUBRANGE_NOT_FOUND_ON_LIB Code: OC 65

Description: A module specifying the lower bound of a subrange is not in the current module list. Therefore, the entire module subrange was not processed.

User Action: Reenter the command, specifying the correct module subrange. To list the current module list, enter a DISPLAY_NEW_LIBRARY command.

--WARNING OC 66-- Duplicate module {text} on file {file}.

Condition Identifier: OCE\$W_DUPLICATE_MODULE_ON_FILE Code: OC 66

Description: The specified file contains a module having the same name as a module processed earlier. The second module is not processed.

User Action: Determine which module (the first or second) is required. Replace the first module with the second module if necessary.

--WARNING OC 67-- Exiting library generator without generating current library.

Condition Identifier: OCE\$W_LIBRARY_NOT_GENERATED Code: OC 67

Description: You ended the session without using the information in the current module list to generate a file.

User Action: None.

--ERROR OC 68-- No modules on current library.

Condition Identifier: OCE\$W_NO_MODULES_ON_CURRENT_LIB Code: OC 68

Description: No modules are displayed because the current module list contains no modules. Note that the module list is cleared after a GENERATE_LIBRARY command is processed.

User Action: Add modules to the module list as needed.

--ERROR OC 69-- Due to errors the generated {text} was placed on {file}.

Condition Identifier: OCE\$W_NEW_FILE_IS_ON_FILE Code: OC 69

Description: The object library generator could not write the library on the specified file so it wrote it on the specified temporary file.

User Action: You can copy the temporary file to another file with the SCL command COPY_FILE.

--ERROR OC 70-- New file was not generated due to errors encountered.

Condition Identifier: OCE\$W_NEW_FILE_NOT_GENERATED Code: OC 70

Description: The object library generator could not generate the file due to errors described in other messages.

User Action: Determine the cause of the errors from the other messages.

--ERROR OC 71-- Entry point {text} already exists - entry point ignored.

Condition Identifier: OCE\$W_XDCL_ALREADY_EXISTS Code: OC 71

Description: The specified entry point name cannot be used because an entry point with that name already exists in the module.

User Action: Reenter the CHANGE_MODULE_ATTRIBUTES command specifying a different new entry point name.

--ERROR OC 72-- Entry point {text} not found - entry point ignored.

Condition Identifier: OCE\$W_XDCL_DOESNT_EXIST Code: OC 72

Description: The specified entry point could not be omitted or have its name changed because the specified entry point could not be found.

User Action: Reenter the CHANGE_MODULE_ATTRIBUTES command specifying the correct entry point names.

--WARNING OC 73-- External reference {text} not satisfied.

Condition Identifier: OCE\$W_EXTERNAL_NOT_SATISFIED Code: OC 73

Description: The specified external reference could not be satisfied from the object libraries specified on the command.

User Action: If a module containing the entry point is to be included in the new object library, enter another SATISFY_EXTERNAL_REFERENCES command specifying an object library containing the entry point that satisfies the unsatisfied external reference.

--ERROR OC 74-- Module {text} has multiple binding sections.

Condition Identifier: OCE\$E_MULTIPLE_BINDING_SECTIONS Code: OC 74

Description: The specified module contains more than one binding section.

User Action: Remove the extra binding section delimiters from the source code and recompile the module.

--WARNING OC 75-- Conflicting length for common block {text}.

Condition Identifier: OCE\$W_CONFLICTING_COMMON_LENGTH Code: OC 75

Description: Module format error. Definitions within two modules for the same common block differ in the length specified.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--WARNING OC 76-- Conflicting attributes for common block {text}.

Condition Identifier: OCE\$W_CONFLICTING_COM_ATTRIBUTE Code: OC 76

Description: Module format error. Definitions within two modules for the same common block differ in the attributes specified.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 77-- Bad debug symbol table in module {text} - symbol number {text}.

Condition Identifier: OCE\$E_BAD_SYMBOL_TABLE Code: OC 77

Description: Module format error. The format of a symbol table record in the module is incorrect.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 78-- Bad line table in module {text}.

Condition Identifier: OCE\$E_BAD_LINE_TABLE Code: OC 78

Description: Module format error. The format of a line table record in the module is incorrect.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 79-- Zero section alignment in module {text}.

Condition Identifier: OCE\$E_ZERO_ALLOCATION_ALIGN Code: OC 79

Description: Module format error. The section alignment value in a section definition within the module is zero.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--WARNING OC 80-- Library {text} not found.

Condition Identifier: OCE\$W_LIBRARY_NOT_FOUND Code: OC 80

Description: The specified object library could not be found.

User Action: Determine if the object library is required for correct processing of the command. If so, reenter the command, specifying the object library correctly.

--ERROR OC 81-- Fatal error in library_generator.

Condition Identifier: OCE\$E_FATAL_ERROR Code: OC 81

Description: The object library generator terminated due to an error in its processing.

User Action: Provide a site analyst with documentation of the error.

--ERROR OC 82-- Insufficient memory for completion of library generation.

Condition Identifier: OCE\$E_INSUFFICIENT_MEMORY Code: OC 82

Description: The object library generator could not complete its processing because not enough memory was available to it.

User Action: Either decrease the size of the object library to be generated or attempt the library generation when more memory is available. If the error recurs, inform a site analyst.

--ERROR OC 83-- Can not compare an object file and an object library.

Condition Identifier: OCE\$E_CANT_COMPARE_FILE_AND_LIB Code: OC 83

Description: The COMPARE_OBJECT_LIBRARIES command cannot compare an object library and an object file. It can compare two object libraries or two object files.

User Action: If desired, you can either generate an object library using the contents of the object file or generate an object file from the contents of the object library (if all modules on the object library are load modules). You could then compare the two object files or two object libraries.

--ERROR OC 84-- Command terminated - new library not generated.

Condition Identifier: OCE\$E_GENERATE_TERMINATED Code: OC 84

Description: The command terminated without successfully generating the file it was to write.

User Action: Determine the reason for the termination from the other messages returned.

--ERROR OC 85-- Module {text} is not an scl procedure.

Condition Identifier: OCE\$E_MODULE_IS_NOT_AN_SCL_PROC Code: OC 85

Description: To generate a procedure text file, all modules in the module list must be SCL procedures.

User Action: Either delete the specified module from the module list or change the value of the FORMAT parameter on the GENERATE_LIBRARY command.

--ERROR OC 86-- Unable to generate a unique local file name.

Condition Identifier: OCE\$E_NO_UNIQUE_FILE_NAME Code: OC 86

Description: Internal object library generator error.

User Action: Provide a site analyst with appropriate documentation of the error.

--ERROR OC 87-- Some of the modules specified could not be {text}.

Condition Identifier: OCE\$E_SOME_MODULES_NOT Code: OC 87

Description: As described in other messages, the object library generator could not process all modules specified on the command.

User Action: Reenter the command, specifying correctly the modules that were not processed.

--ERROR OC 88-- Some of the attributes specified could not be changed.

Condition Identifier: OCE\$E_SOME_ATTRIBUTES_NOT Code: OC 88

Description: As described in other messages, the object library generator could not perform all attribute changes requested by the CHANGE_MODULE_ATTRIBUTES command.

User Action: Reenter the command, specifying correctly the attribute changes that were not performed.

--ERROR OC 89-- Command completed - Non fatal errors encountered.

Condition Identifier: OCE\$W_ERROR_IN_COMMAND Code: OC 89

Description: Although the command completed processing successfully, one or more nonfatal errors occurred. The other messages returned list causes of the nonfatal errors.

User Action: Determine if the nonfatal errors prevented accomplishment of your processing.

--INFORMATIVE OC 90-- {text}

Condition Identifier: OCE\$I_DISPLAY_TOGGLE Code: OC 90

Description: This line informs you of the default display options currently in effect.

User Action: If desired, you can change the default display options with the SET_DISPLAY_OPTIONS command.

--ERROR OC 91-- Module {text} not found after module {text} - Subrange ignored.

Condition Identifier: OCE\$E_RANGE_MODULE_2_NOT_FOUND Code: OC 91

Description: A module specifying the upper bound of a subrange is not in the current module list. Therefore, the entire module subrange was not processed.

User Action: Reenter the command, specifying the correct module subrange. To list the current module list, enter a DISPLAY_NEW_LIBRARY command.

--ERROR OC 92-- Invalid size encountered on object text descriptor in module {text}.

Condition Identifier: OCE\$E_BAD_OTD_SIZE Code: OC 92

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--WARNING OC 93-- Duplicate entry point {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_ENT_PNT_ON_LIB Code: OC 93

Description: An object library specified on the command contains two entry points with the same name.

User Action: Generate a new object library, removing one of the duplicate entry points.

--ERROR OC 94-- Relocation value not found in module {text}.

Condition Identifier: OCE\$RELOCATION_VALUE_NOT_FOUND Code: OC 94

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--WARNING OC 95-- Declaration mismatch on reference to entry point {text} in module {text} from module {text}, using source text checking.

Condition Identifier: OCE\$W_DECLARATION_MISMATCH Code: OC 95

Description: The CREATE_OBJECT_LIBRARY utility detected a possible mismatch between the entry point declaration in one module and its external reference in another module.

User Action: Compare the source text for the two modules to ensure that the entry point and external reference match. Depending on the difference between the entry point and the external reference declarations, you may have to recompile your program, or you may be able to execute the program from the generated library by setting TERMINATION_ERROR_LEVEL=FATAL.

--WARNING OC 96-- WARNING - Nonfatal error(s) encountered during generation of new library.

Condition Identifier: OCE\$W_NON_FATAL_GENERATE_ERROR Code: OC 96

Description: As described by other messages, errors occurred during library generation that did not prevent generation of the library.

User Action: Determine if the errors that occurred are significant.

--INFORMATIVE OC 97-- {text}

Condition Identifier: OCE\$MISC_EXCEPTION Code: OC 97

Description: This is an informative message only.

User Action: None.

--ERROR OC 98-- The bound module has overflowed a {text}.

Condition Identifier: OCE\$CONTAINER_OVERFLOW Code: OC 98

Description: There is too much code to be bound into one module.

User Action: Reduce the size of the module being generated by breaking it into separate bound modules. These separate modules can then be placed on an object library.

--ERROR OC 99-- Module name can not be changed when multiple modules specified.

Condition Identifier: OCE\$E_CANT_CHANGE_MODULE_NAME Code: OC 99

Description: Since more than one module was specified on the CHANGE_MODULE_ATTRIBUTE command, the NEW_NAME parameter is invalid.

User Action: Specify only the module whose name is to be changed for the MODULE parameter on CHANGE_MODULE_ATTRIBUTES.

--ERROR OC 100-- Too many commands on library.

Condition Identifier: OCE\$E_TOO_MANY_COMMANDS_ON_LIB Code: OC 100

Description: The number of commands on the library being generated exceeds the maximum allowed in the command dictionary. The maximum number of commands that are allowed is 7FFFFFFF(16).

User Action: Remove one or more commands from the library and reenter the GENERATE_LIBRARY command.

--ERROR OC 101-- Too many functions on library.

Condition Identifier: OCE\$E_TOO_MANY_FUNCTIONS_ON_LIB Code: OC 101

Description: The number of functions of the library being generated exceeds the maximum allowed in the help module dictionary. The maximum number of functions that are allowed is 7FFFFFFF(16).

User Action: Remove one or more functions from the library and reenter the GENERATE_LIBRARY command.

--ERROR OC 102-- Too many help modules on library.

Condition Identifier: OCE\$E_TOO_MANY_HELP_MODS_ON_LIB Code: OC 102

Description: The number of help modules on the library being generated exceeds the maximum allowed in the help module dictionary. The maximum number of help modules that are allowed is 7FFFFFFF(16).

User Action: Remove one or more help modules from the library and reenter the GENERATE_LIBRARY command.

--ERROR OC 103-- Too many message modules on library.

Condition Identifier: OCE\$E_TOO_MANY_MSG_MODS_ON_LIB Code: OC 103

Description: The number of message modules on the library being generated exceeds the maximum allowed in the message module dictionary. The maximum number of message modules that are allowed is 16,777,216.

User Action: Remove one or more message modules from the library and reenter the GENERATE_LIBRARY command.

--ERROR OC 104-- Too many panels on library.

Condition Identifier: OCE\$E_TOO_MANY_PANELS_ON_LIB Code: OC 104

Description: The number of panel modules on the library being generated exceeds the maximum allowed in the panel module dictionary. The maximum number of panel modules is 16,777,216.

User Action: Remove one or more of the panel modules from the library and reenter the GENERATE_LIBRARY command.

--WARNING OC 105-- Duplicate command {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_COMMANDS_ON_LIB Code: OC 105

Description: A duplicate command was encountered during the library generation.

User Action: One of the duplicate commands or aliases should be deleted or changed.

--WARNING OC 106-- Duplicate function {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_FUNCTIONS_ON_LIB Code: OC 106

Description: A duplicate function was encountered during library generation.

User Action: One of the duplicate functions should be deleted or changed.

--WARNING OC 107-- Duplicate help modules {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_HELP_MODULES_ON_LIB Code: OC 107

Description: A duplicate help module was encountered during library generation.

User Action: One of the duplicate help modules should be deleted or changed.

--WARNING OC 108-- Duplicate message modules {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_MSG_MODULES_ON_LIB Code: OC 108

Description: A duplicate message module was encountered during library generation.

User Action: One of the duplicate message modules should be deleted or changed.

--WARNING OC 109-- Duplicate command {text} in library may cause unpredictable results.

Condition Identifier: OCE\$W_DUP_PANELS_ON_LIB Code: OC 109

Description: A duplicate panel module was encountered during library generation.

User Action: One of the duplicate panel modules should be deleted or changed.

--ERROR OC 110-- {text} must be used alone for parameter {text}.

Condition Identifier: OCE\$E_OPTION_MUST_BE_USED_ALONE Code: OC 110

Description: The specified option can not be used with any other values for the specified parameters.

User Action: Re-enter the option by itself.

--ERROR OC 111-- Modules {text} and {text} with parameter_number {text} of procedure {text} are incompatible types.

Condition Identifier: OCE\$INVALID_TYPE_MATCHING Code: OC 111

Description: Types of parameters are incompatible.

User Action: Change program source code so parameter types are compatible. Recompile.

--ERROR OC 112-- Modules {text} and {text} with parameter_number {text} of procedure {text} are incompatible kinds.

Condition Identifier: OCE\$INVALID_KIND_MATCHING Code: OC 112

Description: Parameters are of incompatible kinds.

User Action: Change program source code so parameter kinds are compatible. Recompile.

--ERROR OC 113-- Modules {text} and {text} with parameter number {text} of procedure {text} are incompatible modes.

Condition Identifier: OCE\$INVALID_MODE_MATCHING Code: OC 113

Description: Modes of parameters are incompatible.

User Action: Change program source code so parameter modes are compatible. Recompile.

--ERROR OC 114-- {text} has character length mismatch on parameter {text}, actual length = {text} and formal length = {text}.

Condition Identifier: OCE\$BAD_CHAR_LENGTH Code: OC 114

Description: Lengths of parameters are incompatible.

User Action: Change program source code so parameters have compatible lengths. Recompile.

--ERROR OC 115-- {text} parameter number {text} actual length of {text} less than formal length of {text}.

Condition Identifier: OCE\$ACTUAL_LESS_THAN_FORMAL Code: OC 115

Description: Lengths of parameters are incompatible.

User Action: Change program source code so parameters have compatible lengths. Recompile.

--ERROR OC 118-- Parameter mismatch ({text}).

Condition Identifier: OCE\$ENTRY_POINT_MISMATCH Code: OC 118

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 119-- The external record for {text} needed by an actual parameter record in module {text} has not been encountered.

Condition Identifier: OCE\$EXT_NOT_FOUND_FOR_ACTUAL Code: OC 119

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 120-- Procedure {text} in modules {text} and {text} do not have the same number of parameters.

Condition Identifier: OCE\$INVALID_PARAM_FOR_PROC Code: OC 120

Description: Number of parameters in call to procedure is incompatible with the number of parameters for procedure.

User Action: Change program source code so the number of parameters are compatible. Recompile.

--ERROR OC 121-- Entry ({text}) not found for formal ({text}).

Condition Identifier: OCE\$ENTRY_NOT_FOUND_FOR_FORMAL Code: OC 121

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 122-- Unallocated common block has invalid section kind in module {text}.

Condition Identifier: OCE\$INVALID_UNALLOC_COMMON_BL Code: OC 122

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 123-- Form definition found in module {text}.

Condition Identifier: OCE\$E_FORM_DEF_FOUND_IN_MODULE Code: OC 123

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 124-- Value of parameter {text} must be FILE or STRING.

Condition Identifier: OCE\$E_MUST_BE_FILE_OR_STRING Code: OC 124

Description: The parameter value must be supplied as a file name or a string. A string must be enclosed in apostrophes (').

User Action: Re-enter command and specify either a file name or a string for the parameter.

--ERROR OC 125-- Only a system or application administrator may specify an application identifier.

Condition Identifier: OCE\$NOT_SYSTEM_OR_APPLIC_ADMIN Code: OC 125

Description: Specifying an application identifier can only be done by a user who is validated to have SYSTEM_ADMINISTRATOR or APPLICATION_ADMINISTRATOR capabilities.

User Action: Do not specify an application identifier on this command.

--ERROR OC 126-- {text} cannot be an application.

Condition Identifier: OCE\$CANNOT_BE_AN_APPLICATION Code: OC 126

Description: Only program descriptions and SCL procedures can be identified as applications.

User Action: If you are trying to identify a load module as an application, you may create a program description for it and specify the application identifier on the program description.

--ERROR OC 127-- Call from {text} to {text} parameter number {text} has an array mismatch.

Condition Identifier: OCE\$FORTRAN_ARRAY_TYPE_MISMATCH Code: OC 127

Description: Array types specified as parameters are incompatible.

User Action: Change program source so array types of parameters are compatible. Recompile.

--ERROR OC 128-- Module {text} is not a program description.

Condition Identifier: OCE\$E_MODULE_IS_NOT_A_PROG_DESC Code: OC 128

Description: Change_program_description cannot be issued for a module that is not a program description.

User Action: Reenter the command specifying a program description.

**--WARNING OC 130-- Entry point {text}, specified on a DELETE_
DECLARATION_MATCHING subcommand, is not in the linked symbol table.**

Condition Identifier: OCE\$W_NAME_NOT_IN_SYMBOL_TABLE Code: OC 130

Description: The entry point specified is not in the list of entry points in the linked symbol table.

User Action: Specify the correct entry point name or change the entry point attributes or linker directives so that it will be in the linked symbol table.

**--WARNING OC 131-- DELETE_DECLARATION_MATCHING cannot be done for
entry points that are not included in a segment specified by INCLUDE_
LINKED_SYMBOLS.**

Condition Identifier: OCE\$W_MUST_INCLUDE_SYMBOLS Code: OC 131

Description: A DELETE_DECLARATION_MATCHING subcommand was specified, but an INCLUDE_LINKED_SYMBOLS subcommand was not.

User Action: In order to delete declaration matching, you must also specify the INCLUDE_LINKED_SYMBOLS subcommand before generating the virtual memory.

**--ERROR OC 132-- Modules {text} and {text} with parameter number {text} of
procedure {text} are incompatible integer lengths.**

Condition Identifier: OCE\$BAD_INTEGER_LENGTH Code: OC 132

Description: Lengths of parameters are incompatible.

User Action: Change program source code so parameters have compatible lengths. Recompile.

**--WARNING OC 133-- No entry points generated by CYBIL were found in
module {text}. The value of attribute CYBIL_PARAMETER_CHECKING was not
changed.**

Condition Identifier: OCE\$W_PARAM_CHECKING_NOT_CYBIL Code: OC 133

Description: The CYBIL_PARAMETER_CHECKING parameter was specified, but no entry points generated by CYBIL were found in the specified module.

User Action: Be sure that you specified the correct module name on the CHANGE_MODULE_ATTRIBUTE command.

**--FATAL OC 134-- Declaration mismatch on reference to entry point {text} in
module {text} from module {text}, using object text checking.**

Condition Identifier: OCE\$F_DECLARATION_MISMATCH Code: OC 134

Description: The CREATE_OBJECT_LIBRARY utility detected a structural mismatch between an entry point declaration and its external reference.

User Action: Be sure that you specified the correct types of parameters in the correct order. Recompile your program, using the same type declarations that were used when compiling the module containing the entry point which caused the declaration mismatch.

--ERROR OC 200-- Missing or empty file: {file}.

Condition Identifier: OCE\$MISSING_OR_EMPTY_FILE Code: OC 200

Description: The object library generator either could not find the file or the file is empty.

User Action: Ensure that the correct file is specified.

--ERROR OC 201-- Missing record or descriptor: {text}.

Condition Identifier: OCE\$MISSING_REC_OR_DESCRIPTOR Code: OC 201

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 202-- Short record or descriptor: {text}.

Condition Identifier: OCE\$SHORT_RECORD_OR_DESCRIPTOR Code: OC 202

Description: A module format error on the CI object module has occurred.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 203-- Identification or library descriptor expected: {text}.

Condition Identifier: OCE\$IDENT_OR_LIB_DESC_EXPECTED Code: OC 203

Description: Library format error.

User Action: Regenerate the object library, if possible. If the error recurs, inform a site analyst.

--ERROR OC 204-- Multiple identification records in same module: {text}.

Condition Identifier: OCE\$MULTIPLE_IDENT_RECORDS Code: OC 204

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 205-- Invalid object text version. {text}.

Condition Identifier: OCE\$INVALID_VERSION Code: OC 205

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 206-- Invalid CPU record kind: {text}.

Condition Identifier: OCE\$INVALID_CPU_RECORD_KIND Code: OC 206

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 207-- Invalid PPU record kind: {text}.

Condition Identifier: OCE\$INVALID_PPU_RECORD_KIND Code: OC 207

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 208-- Invalid object module kind: {text}.

Condition Identifier: OCE\$INVALID_OBJECT_MODULE_KIND Code: OC 208

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 209-- Invalid parameter kind: {text}.

Condition Identifier: OCE\$INVALID_PARAMETER_KIND Code: OC 209

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 210-- Invalid object record kind: {text}.

Condition Identifier: OCE\$INVALID_OBJECT_RECORD_KIND Code: OC 210

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 400-- An object library must be analyzed before it is displayed.

Condition Identifier: OCE\$OBJECT_LIBRARY_NOT_ANALYZED Code: OC 400

Description: There has been no object library analyzed.

User Action: Specify an object library to be analyzed by the USE_LIBRARY subcommand in ANALYZE_OBJECT_LIBRARY.

--ERROR OC 401-- Module {text} is not on the current library.

Condition Identifier: OCE\$UNKNOWN_MODULE Code: OC 401

Description: The module specified is not on the current library.

User Action: Re-enter command and specify correct module name. A list of modules on your library can be obtained by using DISPLAY_OBJECT_LIBRARY L=your_library. If the module you want is on a different library, enter a new USE_LIBRARY command to analyze that library.

--ERROR OC 500-- Non master key-locks {text} and {text} must be equal.

Condition Identifier: OCE\$E_GLOBAL_LOCAL_KEY_MISMATCH Code: OC 500

Description: The key locks must match if they are not the master values.

User Action: Change the key lock values to make EQUAL or MASTER.

--ERROR OC 501-- Internal storage allocation failed - {text}.

Condition Identifier: OCE\$E_STORAGE_ALLOCATION_FAILED Code: OC 501

Description: An internal file has overflowed.

User Action: Inform a site analyst providing appropriate documentation of the error.

--ERROR OC 502-- File {file} is not a library.

Condition Identifier: OCE\$E_FILE_IS_NOT_LIBRARY Code: OC 502

Description: The file specified is not an object library.

User Action: An object library must be generated from the object file using the CREATE_OBJECT_LIBRARY utility.

--ERROR OC 503-- File {file} does not contain a symbol table.

Condition Identifier: OCE\$E_FILE_IS_NOT_SYMBOL_TABLE Code: OC 503

Description: The file specified does not contain a valid linked symbol table.

User Action: The file should be regenerated using LINK_VIRTUAL_ENVIRONMENT.

--ERROR OC 504-- Invalid ring brackets specified {text}, {text}, and {text}.

Condition Identifier: OCE\$E_INVALID_RING_BRACKET Code: OC 504

Description: The ring brackets must be in the form R1<=R2<=R3.

User Action: Reissue the command with valid ring brackets.

--ERROR OC 505-- Duplicate module {text} specified.

Condition Identifier: OCE\$E_DUPLICATE_MODULE_NAMED Code: OC 505

Description: The specified module has already been used.

User Action: None.

--ERROR OC 506-- Duplicate file {file} specified.

Condition Identifier: OCE\$E_DUPLICATE_FILE_NAMED Code: OC 506

Description: The specified file has already been used.

User Action: Reissue the command specifying a different file name.

--ERROR OC 507-- Segment attribute conflict - {text} specified.

Condition Identifier: OCE\$E_SEG_ATTRIBUTE_CONFLICT Code: OC 507

Description: The specified combination of segment attributes can not be defined for a segment.

User Action: Reissue the command specifying a valid combination of segment attributes.

--ERROR OC 508-- Duplicate segment number {text} specified.

Condition Identifier: OCE\$E_DUPLICATE_SEGMENT_NAMED Code: OC 508

Description: A duplicate segment number has been specified.

User Action: Reissue the command specifying a different segment number.

--ERROR OC 509-- Duplicate section name {text} specified.

Condition Identifier: OCE\$E_DUPLICATE_SECTION_NAMED Code: OC 509

Description: A duplicate section name has been specified.

User Action: Reissue the command specifying a different section name.

--ERROR OC 510-- Sections {text} and {text} have duplicate attributes specified.

Condition Identifier: OCE\$E_DUPLICATE_SECTION_ATTR Code: OC 510

Description: The two specified sections have the same attributes associated with them.

User Action: Reissue the command specifying which attributes for each section name.

--ERROR OC 511-- Segment number overflow.

Condition Identifier: OCE\$E_SEGMENT_NUMBER_OVERFLOW Code: OC 511

Description: The linker was unable to generate a new unique segment number.

User Action: Decrease the starting segment number or decrease the number of segments and relink.

--ERROR OC 512-- Invalid symbol table version on file {file}.

Condition Identifier: OCE\$E_INVALID_LST_VERSION Code: OC 512

Description: The specified file contains an invalid or old symbol table.

User Action: Regenerate the symbol table using LINK_VIRTUAL_ENVIRONMENT.

--ERROR OC 513-- Bad object text on file {text} - {text}.

Condition Identifier: OCE\$E_BAD_OBJ_TEXT_ON_FILE Code: OC 513

Description: An object file format error has occurred on the specified file.

User Action: Regenerate the file by recompiling its source code. If the problem recurs, inform a site analyst.

--ERROR OC 514-- Bad object text in module {text}, at record {text} - {text}.

Condition Identifier: OCE\$E_BAD_OBJ_TEXT_IN_MODULE Code: OC 514

Description: An object format error has occurred in the specified module.

User Action: Regenerate the module by recompiling the source code. If the problem recurs, inform the site analyst.

--ERROR OC 515-- Starting procedure {text} not found.

Condition Identifier: OCE\$E_STARTING_PROC_NOT_FOUND Code: OC 515

Description: The specified starting procedure could not be found in the object file and object libraries included in the link.

User Action: Specify an existing entry point or include the file containing the starting procedure in the link.

--ERROR OC 516-- GATE_RING_LEVEL can only be specified when LINKED_SYMBOLS is GATE.

Condition Identifier: OCE\$E_LINKED_SYMBOLS_NOT_GATE Code: OC 516

Description: The GATE_RING_LEVEL parameter was specified when the LINKED_SYMBOLS parameter was not GATE.

User Action: Specify GATE on the LINKED_SYMBOLS parameter or omit the GATE_RING_LEVEL parameter.

--ERROR OC 517-- Both SECTION and SEGMENT_NUMBER specified.

Condition Identifier: OCE\$E_SECTION_SEGMENT_TOGETHER Code: OC 517

Description: Both SECTION and SEGMENT_NUMBER parameters were specified on the command.

User Action: Reissue the command specifying only the SECTION or the SEGMENT_NUMBER parameter.

--ERROR OC 518-- SECTION or SEGMENT_NUMBER must be specified.

Condition Identifier: OCE\$E_MISSING_SECTION_SEGMENT Code: OC 518

Description: Either the SECTION or the SEGMENT_NUMBER parameter must be specified.

User Action: Reissue the command specifying the SECTION or SEGMENT_NUMBER command.

--ERROR OC 519-- SECTION or SEGMENT not found for pointer {text}.

Condition Identifier: OCE\$E_SECTION_OR_SEG_NOT_FOUND Code: OC 519

Description: The section or segment pointed to by the specified variable has not been defined.

User Action: Define the section or segment and relink.

--WARNING OC 600-- Module {text} not included in link.

Condition Identifier: OCE\$W_MODULE_NOT_INCLUDED Code: OC 600

Description: Because of an error in the specified module, the module was not included in the link.

User Action: See link map for the error, correct it and relink.

--WARNING OC 601-- Unsatisfied reference to module {text}.

Condition Identifier: OCE\$W_UNSATISFIED_MODULE Code: OC 601

Description: The specified module could not be found on any of the included object libraries or files.

User Action: Include an object library or file which contains the module or delete the reference to it and relink.

--WARNING OC 602-- Unsatisfied reference to external {text}.

Condition Identifier: OCE\$W_UNSATISFIED_EXTERNAL Code: OC 602

Description: An external reference by one of the modules of the link could not be found in the included object libraries and files.

User Action: Include an object library or file which contains the declaration for the external and relink.

--WARNING OC 603-- No modules on file {file}.

Condition Identifier: OCE\$W_NO_MODULES_ON_FILE Code: OC 603

Description: The specified file contains no modules.

User Action: None.

--WARNING OC 604-- Required library {text} missing.

Condition Identifier: OCE\$W_REQUIRED_LIBRARY_MISSING Code: OC 604

Description: A text-embedded library was specified which has not been specified by a USE_OBJECT_LIBRARY command.

User Action: Specify the library in a USE_OBJECT_LIBRARY command and relink.

--WARNING OC 605-- Attempted to place data in module {text}'s binding section.

Condition Identifier: OCE\$W_DATA_IN_BINDING Code: OC 605

Description: An attempt to place data into a binding section was detected.

User Action: The module's source must be corrected and the module recompiled.

--WARNING OC 606-- Conflicting attributes for common block {text} on module {text}.

Condition Identifier: OCE\$W_CONFLICTING_COMMON_ATTR Code: OC 606

Description: The attributes for the specified common block do not match the first common block defined.

User Action: The module's source must be corrected and recompiled.

--WARNING OC 607-- Conflicting lengths for common block {text} on module {text}.

Condition Identifier: OCE\$W_CONFLICTING_COMMON_LNGTH Code: OC 607

Description: The length for the specified common block does not match the first common block defined.

User Action: The module's source must be corrected and recompiled.

--WARNING OC 608-- Duplicate entry point {text} encountered.

Condition Identifier: OCE\$W_DUPLICATE_ENTRY_POINTS Code: OC 608

Description: A duplicate entry point was found during the link.

User Action: One of the entry points should be deleted.

--FATAL OC 609-- Parameter verification error on external {text} referenced by module {text}, using object text checking.

Condition Identifier: OCE\$F_EXT_PARAM_VERIFICATION Code: OC 609

Description: The virtual environment linker detected a structural mismatch between an entry point declaration and its external reference.

User Action: Be sure you specified the correct types of parameters in the correct order. Recompile your program, using the same type declarations that were used when compiling the module containing the entry point which caused the declaration mismatch.

--WARNING OC 610-- No starting procedure encountered.

Condition Identifier: OCE\$W_NO_STARTING_PROCEDURE Code: OC 610

Description: No starting procedure was found during the link.

User Action: Either change the source to specify a starting procedure and recompile the module, or specify a starting procedure on the SET_LINK_OPTIONS parameter.

--WARNING OC 611-- Possible ring violation with external {text}.

Condition Identifier: OCE\$W_RING_VIOLATION Code: OC 611

Description: A possible ring violation with the specified external was detected.

User Action: Change the ring brackets of the modules and relink.

--WARNING OC 612-- Empty symbol table {text}.

Condition Identifier: OCE\$W_EMPTY_SYMBOL_TABLE Code: OC 612

Description: The specified symbol table is empty.

User Action: Regenerate the symbol table using LINK_VIRTUAL_ENVIRONMENT.

--WARNING OC 613-- Exiting VE Linker without executing the GENERATE command.

Condition Identifier: OCE\$W_GENERATOR_NOT_EXECUTED Code: OC 613

Description: Several link directives were entered but the QUIT command was issued before a GENERATE_VIRTUAL_MEMORY command was issued. The link session was lost.

User Action: Relink and use a GENERATE_VIRTUAL_MEMORY command.

--ERROR OC 614-- Symbol table {file} has not been referenced by an INCLUDE_LINKED_SYMBOLS command.

Condition Identifier: OCE\$E_SYMBOL_TABLE_NOT_DEFINED Code: OC 614

Description: The specified symbol table has not been referenced by an INCLUDE_LINKED_SYMBOLS command.

User Action: Relink using an INCLUDE_LINKED_SYMBOLS command for the specified symbol table.

--ERROR OC 615-- Section {text} was not specified in the section_names parameter of a DEFINE_SEGMENT subcommand.

Condition Identifier: OCE\$E_SEG_NOT_DEFINED_FOR_SECT Code: OC 615

Description: A section name was encountered that was not specified on a DEFINE_SEGMENT subcommand.

User Action: Either define the section name on a DEFINE_SEGMENT subcommand or use the SET_LINK_OPTIONS subcommand to specify CREATE_ONLY_PREDEFINED_SEGMENTS=FALSE.

--INFORMATIVE OC 675-- {text}.

Condition Identifier: OCE\$I_GENERATE_STATUS Code: OC 675

Description: This is an informative message saying no errors or warnings occurred during the GENERATE_VIRTUAL_MEMORY command.

User Action: None.

--WARNING OC 676-- {text}.

Condition Identifier: OCE\$W_GENERATE_STATUS Code: OC 676

Description: Warning message indicating a non-fatal warning occurred during GENERATE_VIRTUAL_MEMORY.

User Action: See the link map and correct the warnings.

--ERROR OC 677-- {text}.

Condition Identifier: OCE\$E_GENERATE_STATUS Code: OC 677

Description: Errors occurred during GENERATE_VIRTUAL_MEMORY.

User Action: See the link map to correct the errors.

--ERROR OC 678-- Invalid debug table version on file {file}.

Condition Identifier: OCE\$E_INVALID_DEBUG_TBL_VERSION Code: OC 678

Description: The specified file contained an old or invalid debug table.

User Action: Regenerate the debug table using LINK_VIRTUAL_MEMORY.

--WARNING OC 679-- Parameter verification on external {text} referenced by module {text}, using source text checking.

Condition Identifier: OCE\$W_EXT_PARAM_VERIFICATION Code: OC 679

Description: The virtual environment linker detected a possible difference between the entry point declaration and the external reference from the specified module.

User Action: Compare the source text for the two modules to ensure that the entry point and external reference match. Depending on the difference between the entry point and external reference declarations, you may have to recompile your program, or you may be able to execute the prelinked module from the generated library by setting TERMINATION_ERROR_LEVEL=FATAL.

--WARNING OC 680-- No build level has been specified for initialization.

Condition Identifier: OCE\$W_BUILD_LEVEL_NOT_SPECIFIED Code: OC 680

Description: A build level must be specified before it can be used to initiate a build level variable.

User Action: Specify the build level on the SET_LINK_OPTIONS command.

--ERROR OC 685-- Add undefined 68000 seq.

Condition Identifier: OCE\$E_ADD_UNDEFINED_68000_SEQ Code: OC 685

Description: An M68000 section could not be assigned to a segment with the correct attributes.

User Action: Define a segment for the section with the correct attributes.

--ERROR OC 687-- Segment {text} overflow.

Condition Identifier: OCE\$E_SEC_OVERFLOW_IN_SEGMENT Code: OC 687

Description: The specified segment could not contain any more information.

User Action: Extend the length of the specified segment.

--ERROR OC 688-- Premature end of file on {text}.

Condition Identifier: OCE\$E_EOF_ON_DEBUG_FILE Code: OC 688

Description: The debug table on the specified file is invalid.

User Action: Regenerate the debug table.

--ERROR OC 689-- A debug table must be specified before it's used.

Condition Identifier: OCE\$E_DEBUG_TABLE_NOT_OPEN Code: OC 689

Description: An attempt was made to access a debug table before it was opened.

User Action: Open the debug table before accessing it.

--ERROR OC 690-- Module {text} not found.

Condition Identifier: OCE\$E_MODULE_ITEM_NOT_FOUND Code: OC 690

Description: The specified module could not be found in the debug table.

User Action: Check the spelling of the module name, and verify that you are using the correct debug table.

--ERROR OC 691-- Address {text} {text} not found.

Condition Identifier: OCE\$E_ADDRESS_NOT_FOUND Code: OC 691

Description: The specified address could not be found in the debug table.

User Action: Verify that the address is correct, and that you are using the correct debug table.

--ERROR OC 692-- Entry point {text} not found.

Condition Identifier: OCE\$E_ENTRY_POINT_NOT_FOUND Code: OC 692

Description: The specified entry point could not be found in the debug table.

User Action: Verify the spelling of the entry point, and make sure that you are using the correct debug table.

--ERROR OC 693-- Invalid address {text} {text} specified.

Condition Identifier: OCE\$E_INVALID_ADDRESS_SPECIFIED Code: OC 693

Description: The address specified is invalid.

User Action: Re-enter the address as a hex number with the segment followed by an 8 hex digit offset.

--ERROR OC 694-- Segment less than minimum pre-linked segment.

Condition Identifier: OCE\$E_SEGMENT_NUMBER_UNDERFLOW Code: OC 694

Description: Segment number specified is less than minimum prelinked segment. The operating system reserves segments 36 through 63 for prelinked programs.

User Action: Re-enter command with valid segment number.

--ERROR OC 695-- Module {text} compiled with OPT=DEBUG - System may abort.

Condition Identifier: OCE\$W_MODULE_COMPILED_OPT_DEBUG Code: OC 695

Description: System may abort when modules are compiled with OPT=DEBUG.

User Action: Recompile module with DA=NONE and OPT=HIGH.

--ERROR OC 696-- Module {text} contains an ALLOCATE or FREE which didn't specify a system heap.

Condition Identifier: OCE\$W_DEFAULT_HEAP_IN_SYSTEM Code: OC 696

Description: A module contains an ALLOCATE or FREE which didn't specify a system heap. You can not use the default heap.

User Action: Change source code to specify the appropriate system heap. Recompile.

--ERROR OC 700-- Inconsistent build options - {text}.

Condition Identifier: OCE\$E_BUILD_OPTION_ERROR Code: OC 700

Description: Inconsistent options were selected with the SET_BUILD_OPTIONS command.

User Action: Reissue the SET_BUILD_OPTIONS command with consistent options.

--ERROR OC 701-- Segment file {text} contains no segment descriptors.

Condition Identifier: OCE\$E_EMPTY_SEGMENT_FILE Code: OC 701

Description: The specified file contains no segment descriptors.

User Action: Regenerate the file using LINK_VIRTUAL_ENVIRONMENT.

--ERROR OC 702-- Segment {text} not found.

Condition Identifier: OCE\$E_EXCH_SEGMENT_NOT_FOUND Code: OC 702

Description: The specified segment number could not be found.

User Action: Check the link and build maps for segment numbers and issue a command with a defined segment number.

--ERROR OC 703-- Segment number {text} greater than segment table size.

Condition Identifier: OCE\$E_SEGMENT_NUMBER_TO_LARGE Code: OC 703

Description: A segment number is greater than the segment table size.

User Action: Increase the segment table size and relink or change the segment number.

--ERROR OC 704-- Duplicate segment {text} specified.

Condition Identifier: OCE\$E_DUPLICATE_SEGMENT_NUMBERS Code: OC 704

Description: A duplicate segment number was detected in the address space.

User Action: Change one of the segment numbers and rebuild.

--ERROR OC 705-- Real memory file overflow.

Condition Identifier: OCE\$E_REAL_MEMORY_OVERFLOW Code: OC 705

Description: A segment was loaded past the end of the real memory.

User Action: Decrease the number of segments or increase the size of the real memory.

--ERROR OC 706-- Segment {text} not found.

Condition Identifier: OCE\$E_SEGMENT_NOT_FOUND Code: OC 706

Description: The specified segment could not be found.

User Action: Check the build and link maps for segment numbers and use an existing segment or define a new segment.

--ERROR OC 707-- {text} extends beyond end of segment {text}.

Condition Identifier: OCE\$E_REAL_MEMORY_SEG_OVERFLOW Code: OC 707

Description: The specified address points outside the segment.

User Action: Change the address and rebuild.

--ERROR OC 708-- Unknown contents on input file {text}.

Condition Identifier: OCE\$E_BAD_INPUT_FILE Code: OC 708

Description: The specified file contains unknown data.

User Action: Regenerate the file using LINK_VIRTUAL_ENVIRONMENT.

--ERROR OC 709-- Tried reading beyond end of real memory image.

Condition Identifier: OCE\$E_REAL_MEMORY_READ_ERROR Code: OC 709

Description: An attempt was made to read past the end of the real memory.

User Action: Change the commands to correct the read.

--ERROR OC 710-- Page width too narrow to display memory.

Condition Identifier: OCE\$E_PAGE_WIDTH_ERROR Code: OC 710

Description: The page width of the output file is too narrow to use the DISPLAY_MEMORY command.

User Action: Increase page width of output file.

--ERROR OC 711-- Invalid address space id.

Condition Identifier: OCE\$E_INVALID_ADDRESS_SPACE_ID Code: OC 711

Description: An invalid address space identifier was specified.

User Action: Issue a command with a valid address space identifier.

--ERROR OC 712-- All real memory pages have been assigned.

Condition Identifier: OCE\$E_NO_MEMORY_AVAILABLE Code: OC 712

Description: The size of real memory is less than the size of the segments being loaded.

User Action: Increase the size of the real memory.

--ERROR OC 713-- Contiguous real memory unavailable for segment {text}.

Condition Identifier: OCE\$E_NO_CONTIGUOUS_REAL_MEMORY Code: OC 713

Description: No contiguous pages remain for the specified segment.

User Action: Increase real memory size or allocate noncontiguous pages or change when pages are loaded.

--ERROR OC 714-- Page at address {text} has been allocated.

Condition Identifier: OCE\$E_PAGE_ALREADY_ALLOCATED Code: OC 714

Description: A page has already been assigned to the specified address.

User Action: Specify a different address or change it when the pages are loaded.

--ERROR OC 715-- Entry {text} not found in symbol table.

Condition Identifier: OCE\$E_EXCHANGE_SYMBOL_NOT_FOUND Code: OC 715

Description: The specified entry point could not be found in the symbol tables.

User Action: Specify a different entry point or check the link map to see where the entry point has been defined. Correct and relink.

--ERROR OC 716-- Asid {text} has already been assigned.

Condition Identifier: OCE\$E_ASID_ALREADY_ASSIGNED Code: OC 716

Description: The specified active segment identifier has already been assigned.

User Action: Choose a different ASID, assign it sooner, or rebuild.

--ERROR OC 717-- Unable to make page table entry.

Condition Identifier: OCE\$E_PAGE_TABLE_RETRY_FAILED Code: OC 717

Description: The page could not be hashed into the page table.

User Action: Expand the page table size and rebuild.

--ERROR OC 718-- Segment {text} load address less than load offset.

Condition Identifier: OCE\$E_INVALID_DEFINE_SEG_ADDR Code: OC 718

Description: The specified segment's load address must be greater or equal to the load offset.

User Action: Change the segment's load address or the load offset.

--ERROR OC 719-- Asid {text} was previously specified.

Condition Identifier: OCE\$E_DUPLICATE_ASID_SPECIFIED Code: OC 719

Description: The specified active segment identifier has already been specified.

User Action: Change one of the ASIDs so that they are both unique.

--ERROR OC 720-- Page Table overflow.

Condition Identifier: OCE\$E_PAGE_TABLE_OVERFLOW Code: OC 720

Description: The page table contains less entries than there are pages.

User Action: Increase page size or increase page table size and rebuild.

--ERROR OC 721-- Invalid ring bracket specified ({text},{text}).

Condition Identifier: OCE\$E_INVALID_RING1_RING2 Code: OC 721

Description: The specified ring brackets must be in the form R1 <= R2.

User Action: Change the ring brackets and rebuild.

--ERROR OC 722-- Invalid file name {text} specified.

Condition Identifier: OCE\$E_INVALID_FILE_NAME Code: OC 722

Description: An invalid file name was specified.

User Action: Correct the file name and rebuild.

--ERROR OC 723-- {text} must be a power of two.

Condition Identifier: OCE\$E_INTEGER_NOT_POWER_OF_TWO Code: OC 723

Description: The specified integer must be a power of two.

User Action: Correct the integer and rebuild.

--ERROR OC 724-- Unable to assign asid {text} to page table.

Condition Identifier: OCE\$E_ASID_WONT_HASH_IN_PT Code: OC 724

Description: The specified active segment identifier did not hash into the page table.

User Action: Change the ASID and rebuild.

--ERROR OC 725-- Exchange Package in segment {text} contains invalid segment table length.

Condition Identifier: OCE\$E_INVALID_XP_SEG_TABLE LENG Code: OC 725

Description: The exchange package in the specified segment contains an invalid segment table length.

User Action: Correct the exchange package, relink and rebuild.

--ERROR OC 726-- Load file {text} contains invalid version number.

Condition Identifier: OCE\$E_INVALID_LOAD_FILE_VERSION Code: OC 726

Description: The specified load file contains old or invalid segment files

User Action: Regenerate the load file using LINK_VIRTUAL_ENVIRONMENT.

--WARNING OC 800-- Exiting Real Memory Builder without executing GENERATE command.

Condition Identifier: OCE\$W_REAL_MEMORY_NOT_GENERATED Code: OC 800

Description: The REAL_MEMORY_BUILDER was issued several directives and a QUIT command without being issued a GENERATE_REAL_MEMORY command.

User Action: Reissue directives followed by a GENERATE_REAL_MEMORY command.

--WARNING OC 801-- {text} undefined and cannot be printed.

Condition Identifier: OCE\$W_NO_EXCHANGE_PACKAGE Code: OC 801

Description: The specified exchange package was not defined and could not be displayed.

User Action: Define the address of the exchange package and then display it.

--WARNING OC 802-- Beginning display memory address is beyond end of real memory image.

Condition Identifier: OCE\$W_INVALID_DISPLAY_MEM_ADDR Code: OC 802

Description: The display memory address must start inside the real memory.

User Action: Specify an address to display which is in real memory.

--FATAL OC 1010-- Module kind {text} invalid.

Condition Identifier: OCE\$INVALID_MODULE_KIND Code: OC 1010

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1020-- No section_definition for section number {text}.

Condition Identifier: OCE\$NO_SECTION_DEFINITION Code: OC 1020

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1050-- No code section found for module {text}.

Condition Identifier: OCE\$NO_CODE_SECTION Code: OC 1050

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--ERROR OC 1060-- {text} object library generated without section_maps.

Condition Identifier: OCE\$NO_SECTION_MAPS Code: OC 1060

Description: A module on the object library is bound and does not have binary section maps. In order to generate a correction for a bound module, the module must have section maps.

User Action: Regenerate the module using the parameter INCLUDE_BINARY_SECTION_MAPS=TRUE on the CREATE_MODULE subcommand.

--FATAL OC 1070-- Relocation record container is invalid.

Condition Identifier: OCE\$INVALID_CONTAINER Code: OC 1070

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1080-- Relocation record address field is invalid.

Condition Identifier: OCE\$INVALID_RELOCATION_ADDRESS Code: OC 1080

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1090-- Record type {text} is unknown.

Condition Identifier: OCE\$UNEXPECTED_RECORD_KIND Code: OC 1090

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1110-- Section map pointer points to middle of {text} record.

Condition Identifier: OCE\$SECTION_MAP_INVALID_POINTER Code: OC 1110

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1130-- Identification record was expected.

Condition Identifier: OCE\$ID_RECORD_EXPECTED Code: OC 1130

Description: Module format error.

User Action: Regenerate the module by recompiling its source code. If the error recurs, inform a site analyst.

--FATAL OC 1170-- Incompatible object library version on file {text}.

Condition Identifier: OCE\$INVALID_LIBRARY_VERSION Code: OC 1170

Description: The object library version as recorded in the library header is incompatible with the current version of the object library generator.

User Action: If possible, regenerate the object library using the original modules.

--FATAL OC 1180-- Unexpected error occurred while {text} during correction generation.

Condition Identifier: OCE\$ERROR_IN_CORRECTION_GEN Code: OC 1180

Description: Internal correction generator error.

User Action: Inform a site analyst of the error with appropriate documentation of the error.

--ERROR OF 1-- A prior operator message has not been received by the task.

Condition Identifier: OFE\$PRIOR_RESPONSE_NOT_CLEARED Code: OF 1

Description: A prior operator message has not been received by this task.

User Action: Resubmit the message.

--ERROR OF 2-- There is no action message posted for job {text}.

Condition Identifier: OFE\$ACTION_MESSAGE_NOT_POSTED Code: OF 2

Description: The attempt to send an operator message requesting a tape mount failed.

User Action: Notify a site analyst.

--ERROR OF 3-- An operator facility message structure error exists.

Condition Identifier: OFE\$ALLOCATE_STRUCTURE_FAILED Code: OF 3

Description: This is an internal error code that is used only when the operator tables are not in proper order.

User Action: Contact the site analyst.

--ERROR OF 5-- The maximum number of operator action messages has been exceeded.

Condition Identifier: OFE\$MAX_JOB_OPERATOR_ACTIONS Code: OF 5

Description: If the request were processed, the number of outstanding requests would exceed the maximum allowed number of operator action requests.

User Action: Resubmit request after one or more of the outstanding requests have been processed.

--ERROR OF 6-- The operator has not acted upon a previous message.

Condition Identifier: OFE\$PREVIOUS_MSG_NOT_CLEARED Code: OF 6

Description: The message cannot be sent because the operator has not acted upon the last message the task sent to the operator.

User Action: Resubmit the message after the previous message is cleared.

--ERROR OF 7-- The operator id, {text}, is invalid.

Condition Identifier: OFE\$INVALID_OPERATOR_ID Code: OF 7

Description: The only valid operator id is for the system operator.

User Action: Correct the operator id.

--ERROR OF 8-- The operator message area was not found.

Condition Identifier: OF\$OPERATOR_MESSAGE_AREA Code: OF 8

Description: Operator message area was not found.

User Action: Contact the site analyst.

--ERROR OF 9-- The operator message string specified is too long.

Condition Identifier: OF\$MESSAGE_TOO_LONG Code: OF 9

Description: The specified operator message is too long.

User Action: Shorten the operator message.

--ERROR OF 10-- The task has not received an operator message.

Condition Identifier: OF\$MESSAGE_NOT_AVAILABLE Code: OF 10

Description: The task has not received an operator message.

User Action: If appropriate, repeat the receive message request until a message is available.

--ERROR OS 10-- The job has active child tasks; {text}

Condition Identifier: OSE\$JOB_HAS_ACTIVE_CHILD_TASKS Code: OS 10

Description: The specified command is not allowed when the job has active child tasks.

User Action: Try again after the child tasks have terminated.

--WARNING OS 11-- Multiprocessing capabilities for job are already on; {text}

Condition Identifier: OSE\$MULTIPROCESSING_ALREADY_ON Code: OS 11

Description: Two SET_MULTIPROCESSING_OPTIONS commands were issued to turn multiprocessing on for the job.

User Action: Remove redundant request.

--ERROR OS 12-- Command was not called by job monitor: {text}

Condition Identifier: OSE\$NOT_CALLED_BY_JOB_MONITOR Code: OS 12

Description: The command issued is only allowed from the job monitor task.

User Action: Reissue the command from the job monitor task.

--ERROR OS 13-- No specific processor was chosen for the job; {text} not executed

Condition Identifier: OSE\$NO_PROCESSOR_SELECTED Code: OS 13

Description: The required parameter that specifies upon which processor the task is to run was not given.

User Action: Reissue the command with the required parameters.

--ERROR OS 14-- The processor is not defined in this configuration; {text} not executed

Condition Identifier: OSE\$PROCESSOR_NOT_DEFINED Code: OS 14

Description: The specified processor is not defined in the current configuration.

User Action: Use the interface PMP\$GET_PROCESSOR_DESCRIPTIONS to get information about the processors that are available in the current configuration, and choose one of them.

--ERROR OS 15-- {text} cannot select a processor that is down or off

Condition Identifier: OSE\$PROCESSOR_NOT_ON Code: OS 15

Description: The requested processor has been turned off or downed.

User Action: Try again after the desired processor becomes available.

--ERROR OS 20-- The system task, {text}, is not defined.

Condition Identifier: OSE\$SYSTEM_TASK_NOT_DEFINED Code: OS 20

Description: The system task with the name you specified is unknown to the system job; that is, it has not yet been defined with the DEFINE_SYSTEM_TASK command. Any attempt to activate, deactivate, or remove an undefined task results in this error.

User Action: Check for correct spelling of defined system task, or perform a DEFINE_SYSTEM_TASK for the named task.

--ERROR OS 21-- The system task, {text}, is already defined.

Condition Identifier: OSE\$SYSTEM_TASK_ALREADY_DEFINED Code: OS 21

Description: The task name you specified has already been used to define and name a system task.

User Action: Specify a different name when defining this task, or issue a REMOVE_SYSTEM_TASK to remove the current definition for this task and reissue the DEFINE_SYSTEM_TASK command.

--ERROR OS 22-- The system task, {text}, is not active.

Condition Identifier: OSE\$SYSTEM_TASK_NOT_ACTIVE Code: OS 22

Description: Either the specified system task was not executing at the time that a DEACTIVATE_SYSTEM_TASK command was issued, or two DEACTIVATE_SYSTEM_TASK commands were issued for the specified system task.

User Action: Determine that the specified system task is active before issuing the command, or remove the redundant request.

--ERROR OS 23-- The system task, {text}, is currently active.

Condition Identifier: OSE\$SYSTEM_TASK_ACTIVE Code: OS 23

Description: The specified system task was executing at the time you issued a REMOVE_SYSTEM_TASK command.

User Action: Issue a DEACTIVATE_SYSTEM_TASK command for the specified system task, then reissue the REMOVE_SYSTEM_TASK command for that task.

--ERROR OS 24-- The system task, {text}, is already enabled.

Condition Identifier: OSE\$SYSTEM_TASK_ENABLED Code: OS 24

Description: Either the specified system task was executing at the time an ACTIVATE_SYSTEM_TASK command was issued, or two ACTIVATE_SYSTEM_TASK commands were issued for the same task.

User Action: Determine that the specified system task is not active before issuing the command, or remove the redundant request.

--ERROR OS 26-- The system task, {text}, cannot be controlled via the command {text}

Condition Identifier: OSE\$TASK_NOT_UNDER_OPER_CONTROL Code: OS 26

Description: The specified task cannot be activated or deactivated by the operator at the system console.

User Action: Do not issue the ACTIVATE_SYSTEM_TASK or DEACTIVATE_SYSTEM_TASK commands.

--ERROR OS 27-- The system task, {text}, was terminated by an IDLE_SYSTEM or a TERMINATE_SYSTEM command.

Condition Identifier: OSE\$TERMINATED_BY_IDLE_SYSTEM Code: OS 27

Description: The specified task was terminated as a result of a condition that caused the system to idle or terminate because the task was defined using the command DEFINE_SYSTEM_TASK with the parameter RESTART_AFTER_IDLE=FALSE. Any task defined in this way will not restart after the system resumes execution.

User Action: If the task is to restart after a system idle, define the task with the parameter RESTART_AFTER_IDLE=TRUE.

--ERROR OS 28-- The system task, {text}, will not complete and therefore its definition cannot be removed.

Condition Identifier: OSE\$SYSTEM_TASK_WONT_COMPLETE Code: OS 28

Description: An attempt was made to deactivate the specified task, but it will not deactivate and terminate, and a subsequent REMOVE_SYSTEM_TASK command has detected this condition.

User Action: Do not issue the REMOVE_SYSTEM_TASK command until a DISPLAY_SYSTEM_TASK_TABLE command shows that the specified system task is complete.

--ERROR OS 50-- Incorrect ost\$wait_activity.

Condition Identifier: OSE\$INCORRECT_ACTIVITY Code: OS 50

Description: The valid activity specifications are for time period expiration, task termination, and receipt of a queue message or for the null activity identifier.

User Action: Specify a valid activity.

--ERROR OS 60-- Unknown condition selector - osp\$set_status_from_condition.

Condition Identifier: OSE\$INVALID_CONDITION_SELECTOR Code: OS 60

Description: The condition must be specified either by its integer condition code or by its condition identifier.

User Action: Correct the condition selector.

--ERROR OS 61-- No system condition specified - osp\$set_status_from_condition.

Condition Identifier: OSE\$EMPTY_SYSTEM_CONDITION Code: OS 61

Description: The valid system conditions are described in the condition processing section.

User Action: Specify a set of one or more system conditions.

--ERROR OS 62-- No block exit reason specified - osp\$set_status_from_condition.

Condition Identifier: OSE\$EMPTY_BLOCK_EXIT_REASON Code: OS 62

Description: The valid block exit reasons are non-local block exit, program termination, or program abort.

User Action: Specify a set of one or more block exit reasons.

--ERROR OS 63-- Undefined segment condition PVA={text} AT P={text}.

Condition Identifier: OSE\$UNKNOWN_SEGMENT_CONDITION Code: OS 63

Description: The valid segment conditions are for a read operation beyond EOI, a read or write operation beyond the maximum segment length, and an I/O error when the system copies the segment to backup storage.

User Action: Specify a valid segment condition identifier.

--ERROR OS 64-- Incorrect save_area specified - osp\$set_status_from_condition.

Condition Identifier: OSE\$INVALID_SAVE_AREA Code: OS 64

Description: The specified save area pointer should be the pointer the system passes to the condition handler when the condition occurred.

User Action: Specify the save area pointer passed to the condition handler.

--ERROR OS 66-- Undefined interactive condition.

Condition Identifier: OSE\$UNKNOWN_INTERACTIVE_COND Code: OS 66

Description: The valid interactive conditions are pause break and terminate break.

User Action: Specify a valid interactive condition identifier.

--ERROR OS 100-- Unrecognizable message level.

Condition Identifier: OSE\$BAD_MESSAGE_LEVEL Code: OS 100

Description: The valid message level specifications are for brief and full message levels or for the current message level.

User Action: Correct the message level specification.

--ERROR OS 101-- Incorrect name for natural language: {text}.

Condition Identifier: OSE\$BAD_NATURAL_LANGUAGE Code: OS 101

Description: The name used to denote a natural language must be a valid SCL name.

User Action: Correct the natural language name.

--ERROR OS 102-- Unrecognizable interaction style.

Condition Identifier: OSE\$BAD_INTERACTION_STYLE Code: OS 102

Description: The valid interaction style specifications are for line and screen.

User Action: Correct the interaction style specification.

--ERROR OS 103-- Incorrect day of week specified.

Condition Identifier: OSE\$BAD_DAY_OF_WEEK Code: OS 103

Description: The valid days of the week are specified by OSC\$MONDAY through OSC\$SUNDAY.

User Action: Correct the day of the week specification.

--ERROR OS 500-- Invalid system default selected.

Condition Identifier: OSE\$INVALID_OS_DEFAULT_SELECTED Code: OS 500

Description: An invalid operating system default was selected.

User Action: Specify a valid system default.

--ERROR OS 801-- The job has an active connection to a Cyber 170 process that is not available in stand alone mode.

Condition Identifier: OSE\$MEM_LINK_NOT_AVAILABLE Code: OS 801

Description: A job had a connection to a CYBER 170 process when NOS/VE was running in dual-state mode. The system was then terminated (or failed) and NOS/VE was brought up in stand-alone mode. An attempt was made to recover the job, but since it required the presence of dual-state, the job could not be recovered completely.

User Action: No user action is possible. The job will be terminated.

--ERROR OS 900-- Address not readable.

Condition Identifier: OSE\$ADDRESS_NOT_READABLE Code: OS 900

Description: The specified address could not be read.

User Action: Specify a valid address.

--ERROR OS 901-- Address not writable.

Condition Identifier: OSE\$ADDRESS_NOT_WRITABLE Code: OS 901

Description: The specified address could not be written.

User Action: Specify a valid address.

--ERROR OS 9999-- ose\$undefined_condition - - The following error was encountered, but no status condition existed to represent it:

***** {text}

Condition Identifier: OSE\$UNDEFINED_CONDITION Code: OS 9999

Description: An error internal to the OS product has occurred.

User Action: Notify your site support group.

--FATAL PA 2-- (PA 0002) Pascal detected {text} error(s), TEL = {text} causes abnormal termination.

Condition Identifier: PA\$PASCAL_ERROR_TERMINATION Code: PA 2

Description: There are errors in the program.

User Action: Check listing file or errors.

--FATAL PA 4-- (PA 0004) INPUT file is \$NULL, no compilation performed.

Condition Identifier: PA\$INPUT_FILE_NULL Code: PA 4

Description: No compilation was performed because the input file is empty.

User Action: Do not attempt to compile and run program when input file is empty.

--FATAL PA 5-- (PA 0005) The FILE_CONTENTS attribute for the INPUT file must be either UNKNOWN or LEGIBLE.

Condition Identifier: PA\$INPUT_FILE_CONTENTS Code: PA 5

Description: Self-explanatory.

User Action: Make the FILE_CONTENTS attribute either UNKNOWN or LEGIBLE.

--FATAL PA 6-- (PA 0006) The FILE_ORGANIZATION attribute for the INPUT file must be SEQUENTIAL.

Condition Identifier: PA\$INPUT_FILE_ORGANIZATION Code: PA 6

Description: Self-explanatory.

User Action: Make the FILE_ORGANIZATION attribute SEQUENTIAL.

--FATAL PA 7-- (PA 0007) The FILE_STRUCTURE attribute for the INPUT file must be UNKNOWN or DATA.

Condition Identifier: PA\$INPUT_FILE_STRUCTURE Code: PA 7

Description: Self-explanatory.

User Action: Make the FILE_STRUCTURE attribute either UNKNOWN or DATA.

--FATAL PA 8-- (PA 0008) The GLOBAL_ACCESS_MODE attribute for the INPUT file must contain READ.

Condition Identifier: PA\$INPUT_GLOBAL_ACCESS_MODE Code: PA 8

Description: Self-explanatory.

User Action: Make sure the GLOBAL_ACCESS_MODE attribute contains READ.

--FATAL PA 9-- (PA 0009) The FILE_CONTENTS attribute for the BINARY file must be either UNKNOWN or OBJECT.

Condition Identifier: PA\$BINARY_FILE_CONTENTS Code: PA 9

Description: Self-explanatory.

User Action: Make the FILE_CONTENTS attribute either UNKNOWN or OBJECT.

--FATAL PA 10-- (PA 0010) The FILE_ORGANIZATION attribute for the BINARY file must be SEQUENTIAL.

Condition Identifier: PA\$BINARY_FILE_ORGANIZATION Code: PA 10

Description: Self-explanatory.

User Action: Make the FILE_ORGNIZATION attribute SEQUENTIAL.

--FATAL PA 11-- (PA 0011) The FILE_STRUCTURE attribute for the BINARY file must be UNKNOWN or DATA.

Condition Identifier: PA\$BINARY_FILE_STRUCTURE Code: PA 11

Description: Self-explanatory.

User Action: Make the FILE_STRUCTURE attribute either UNKNOWN or DATA.

--FATAL PA 12-- (PA 0012) The GLOBAL_ACCESS_MODE attribute for the BINARY file must contain MODIFY.

Condition Identifier: PAE\$BINARY_GLOBAL_ACCESS_MODE Code: PA 12

Description: Self-explanatory.

User Action: Make sure the GLOBAL_ACCESS_MODE attribute contains MODIFY.

--FATAL PA 13-- (PA 0013) The FILE_CONTENTS attribute for the LIST file must be either UNKNOWN or LEGIBLE or LIST.

Condition Identifier: PAE\$LIST_FILE_CONTENTS Code: PA 13

Description: Self-explanatory.

User Action: Make the FILE_CONTENTS attribute either UNKNOWN or LEGIBLE or LIST.

--FATAL PA 14-- (PA 0014) The FILE_ORGANIZATION attribute for the LIST file must be SEQUENTIAL.

Condition Identifier: PAE\$LIST_FILE_ORGANIZATION Code: PA 14

Description: Self-explanatory.

User Action: Make the FILE_ORGANIZATION attribute SEQUENTIAL.

--FATAL PA 15-- (PA 0015) The FILE_STRUCTURE attribute for the LIST file must be UNKNOWN or DATA.

Condition Identifier: PAE\$LIST_FILE_STRUCTURE Code: PA 15

Description: Self-explanatory.

User Action: Make the FILE_STRUCTURE attribute either UNKNOWN or DATA.

--FATAL PA 16-- (PA 0016) The GLOBAL_ACCESS_MODE attribute for the LIST file must contain MODIFY.

Condition Identifier: PAE\$LIST_GLOBAL_ACCESS_MODE Code: PA 16

Description: Self-explanatory.

User Action: Make sure the GLOBAL_ACCESS_MODE attribute contains MODIFY.

--FATAL PA 17-- (PA 0017) The FILE_CONTENTS attribute for the ERROR file must be either UNKNOWN or LEGIBLE or LIST.

Condition Identifier: PAE\$ERROR_FILE_CONTENTS Code: PA 17

Description: Self-explanatory.

User Action: Make the FILE_CONTENTS attribute either UNKNOWN or LEGIBLE or LIST.

--FATAL PA 18-- (PA 0018) The FILE_ORGANIZATION attribute for the ERROR file must be SEQUENTIAL.

Condition Identifier: PAE\$ERROR_FILE_ORGANIZATION Code: PA 18

Description: Self-explanatory.

User Action: Make the FILE_ORGANIZATION attribute SEQUENTIAL.

--FATAL PA 19-- (PA 0019) The FILE_STRUCTURE attribute for the LIST file must be UNKNOWN or DATA.

Condition Identifier: PAE\$ERROR_FILE_STRUCTURE Code: PA 19

Description: Self-explanatory.

User Action: Make the FILE_STRUCTURE attribute either UNKNOWN or DATA.

--FATAL PA 20-- (PA 0020) The GLOBAL_ACCESS_MODE attribute for the ERROR file must contain MODIFY.

Condition Identifier: PAE\$ERROR_GLOBAL_ACCESS_MODE Code: PA 20

Description: Self-explanatory.

User Action: Make sure the GLOBAL_ACCESS_MODE attribute contains MODIFY.

--FATAL PA 21-- (PA 0021) STANDARDS_DIAGNOSTICS first option must be ERROR_LEVEL or NONE.

Condition Identifier: PA\$STAND_DIAG_FIRST_OPTION Code: PA 21

Description: Self-explanatory.

User Action: Make sure the first option is error level or none.

--FATAL PA 22-- (PA 0022) STANDARDS_DIAGNOSTICS second option must be ANSI or ISO.

Condition Identifier: PA\$STAND_DIAG_SECOND_OPTION Code: PA 22

Description: Self-explanatory.

User Action: Make sure the second option is ANSI or ISO.

--FATAL PA 23-- (PA 0023) STANDARDS_DIAGNOSTICS allowed only two options.

Condition Identifier: PA\$STAND_DIAG_EXTRA_OPTION Code: PA 23

Description: Only two options are allowed: error level and standard identifier.

User Action: Make sure there are only two options.

--FATAL PA 24-- (PA 0024) STANDARDS_DIAGNOSTICS option conflict, options other than NONE ignored.

Condition Identifier: PA\$STAND_DIAG_BAD_OPTION Code: PA 24

Description: Either the first or second option is not valid.

User Action: Be sure both the options are valid.

--FATAL PA 25-- (PA 0025) The FILE_CONTENTS attribute for an INCLUDE file must be either UNKNOWN or LEGIBLE.

Condition Identifier: PA\$INCLUDE_FILE_CONTENTS Code: PA 25

Description: Self-explanatory.

User Action: Make the FILE_CONTENTS attribute either unknown or LEGIBLE.

--FATAL PA 26-- (PA 0026) The FILE_ORGANIZATION attribute for an INCLUDE file must be SEQUENTIAL.

Condition Identifier: PA\$INCLUDE_FILE_ORGANIZATION Code: PA 26

Description: Self-explanatory.

User Action: Make the FILE_ORGANIZATION attribute SEQUENTIAL.

--FATAL PA 27-- (PA 0027) The FILE_STRUCTURE attribute for an INCLUDE file must be UNKNOWN or DATA.

Condition Identifier: PA\$INCLUDE_FILE_STRUCTURE Code: PA 27

Description: Self-explanatory.

User Action: Make the FILE_STRUCTURE attribute either UNKNOWN or DATA.

--FATAL PA 28-- (PA 0028) The GLOBAL_ACCESS_MODE attribute for an INCLUDE file must contain READ.

Condition Identifier: PA\$INCLUDE_GLOBAL_ACCESS_MODE Code: PA 28

Description: Self-explanatory.

User Action: Make sure the GLOBAL_ACCESS_MODE attribute contains READ.

--CATASTROPHIC PA 2000-- Internal error. Xref_context_index < 1.

Condition Identifier: PA\$XREF_CONTEXT_INDEX_LESS_ONE Code: PA 2000

Description: This is an internal error.

User Action: Follow site defined procedure for reporting software errors.

--CATASTROPHIC PA 2001-- Internal error. Xref_context_index > 100.

Condition Identifier: PAE\$XREF_INDEX_GTR_THAN_100 Code: PA 2001

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2002-- Internal error. BIFS: unimplemented built_in function or procedure - {text}.

Condition Identifier: PAE\$FUNCTION_UNIMPLEMENTED Code: PA 2002

Description: This is an internal error.

User Action: Follow site defined procedure for reporting software errors.

--CATASTROPHIC PA 2003-- Internal error. Evaluate_integer_constant: Bad integer constant {text}.

Condition Identifier: PAE\$BAD_INTEGER_CONSTANT Code: PA 2003

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2004-- Internal error. Evaluate_real_constant: bad constant {text}.

Condition Identifier: PAE\$BAD_REAL_CONSTANT Code: PA 2004

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2005-- Internal error. Process_declarations: bad word symbol {text}.

Condition Identifier: PAE\$NOT_DECLARATION_WS Code: PA 2005

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2006-- Internal error. SPAN: type not boolean, char, integer, or ordinal.

Condition Identifier: PAE\$INCORRECT_SPAN_TYPE Code: PA 2006

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2007-- Attempt to subscript object known to be not array or string.

Condition Identifier: PAE\$INVALID_SUBSCRIPT Code: PA 2007

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2008-- Internal error. There are too many inserts for message.

Condition Identifier: PAE\$TOO_MANY_INSERTS Code: PA 2008

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2009-- Internal error. Insert_ordinal_value: the type was not an ordinal type.

Condition Identifier: PAE\$TYPE_NOT_ORDINAL Code: PA 2009

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2010-- Internal error. Insert_ordinal_value: the value is not in the range of the type.

Condition Identifier: PAE\$TYPE_RANGE_EXCEEDED Code: PA 2010

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2012-- Internal error. EXPR: unexpected type action.

Condition Identifier: PAE\$UNEXPECTED_TYPE_ACTION Code: PA 2012

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2013-- Internal error. EXPR: unexpected code action.

Condition Identifier: PAE\$UNEXPECTED_CODE_ACTION Code: PA 2013

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2014-- Internal error. Code_range_check: non_ordinal type used.

Condition Identifier: PAE\$CODE_RANGE_CHECK_TYPE Code: PA 2014

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2015-- Internal error. Code_subscript_check: non_ordinal type used.

Condition Identifier: PAE\$CODE_SUBSCRIPT_CHECK_TYPE Code: PA 2015

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2016-- Internal error. Primary: not prepared to cope with token. Token name = {text}.

Condition Identifier: PAE\$PRIMARY_TOKEN_ERROR Code: PA 2016

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2017-- Internal error. Stack_base_addr: asked for lexical level is greater than current lexical level.

Condition Identifier: PAE\$LEXICAL_LEVEL_TOO_LARGE Code: PA 2017

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2018-- Internal error. Create_debug_st_entry: unexpected DT entry kind.

Condition Identifier: PAE\$UNEXPECTED_DT_ENTRY Code: PA 2018

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--CATASTROPHIC PA 2023-- Internal error. Get_file_name called with identifier <> variable or field.

Condition Identifier: PAE\$ILLEGAL_ID_GET_FILE_NAME Code: PA 2023

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 4970-- {text}: Use of the NOENTRY pragma is non-standard.

Condition Identifier: PA\$NOENTRY_PRAGMA Code: PA 4970

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--WARNING PA 4971-- {text}: Use of the ENTRY pragma is non-standard.

Condition Identifier: PA\$ENTRY_PRAGMA Code: PA 4971

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4972-- {text}: Use of predefined constant identifier {text} is non-standard.

Condition Identifier: PA\$CONSTANT_IDENTIFIER_NON_STD Code: PA 4972

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4973-- {text}: Set member iteration FOR statement is non-standard.

Condition Identifier: PA\$SET_MEM_ITERATION_NON_STD Code: PA 4973

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4974-- {text}: Use of set symmetric difference operator is non-standard.

Condition Identifier: PA\$SET_SYMM_DIFF_NON_STD Code: PA 4974

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4975-- {text}: Use of extended integers is non-standard.

Condition Identifier: PA\$RADIX_NON_STD Code: PA 4975

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4976-- {text}: Direct access file type is non-standard.

Condition Identifier: PA\$DIRECT_FILE_TYP_NON_STD Code: PA 4976

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4977-- {text}: Use of direct access file {text} is non-standard.

Condition Identifier: PA\$DIRECT_FILE_NON_STD Code: PA 4977

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4978-- {text}: Use of underscore or \$ in identifier {text} is non-standard.

Condition Identifier: PA\$UNDERSCORE_IN_ID Code: PA 4978

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4979-- {text}: Array constant is non-standard.

Condition Identifier: PA\$ARRAY_CONST_NON_STD Code: PA 4979

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4980-- {text}: Record constant is non-standard.

Condition Identifier: PA\$RECORD_CONST_NON_STD Code: PA 4980

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4981-- {text}: Set constant is non-standard.

Condition Identifier: PA\$SET_CONST_NON_STD Code: PA 4981

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4982-- {text}: Range in case constant list is non-standard.

Condition Identifier: PA\$RANGE_IN_CASE_LIST Code: PA 4982

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4983-- {text}: Substring notation is non-standard.

Condition Identifier: PA\$SUBSTR_NOTATION_NON_STD Code: PA 4983

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4984-- {text}: Use of dynamic strings is non-standard.

Condition Identifier: PA\$DYNAMIC_STR_USAGE_NON_STD Code: PA 4984

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4985-- {text}: Use of variable string type is non-standard.

Condition Identifier: PA\$VAR_STR_TYPE_NON_STD Code: PA 4985

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4986-- {text}: Use of fixed length string with upper bound of 1 is non-standard.

Condition Identifier: PA\$FIX_STR_UPPER_ONE_NON_STD Code: PA 4986

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4987-- {text}: Predefined identifier {text} is non-standard.

Condition Identifier: PA\$PREDEFINED_ID_NON_STD Code: PA 4987

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4988-- {text}: Input of string type from a textfile is non-standard.

Condition Identifier: PA\$STR_TYPE_INPUT_NON_STD Code: PA 4988

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4989-- {text}: Conformant-array parameters are non-standard.

Condition Identifier: PA\$CONFORM_ARRAY_PARAM_NON_STD Code: PA 4989

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4990-- {text}: Use of the null string is non-standard.

Condition Identifier: PAE\$NULL_STR_USAGE_NON_STD Code: PA 4990

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4991-- {text}: OTHERWISE is non-standard.

Condition Identifier: PAE\$OTHERWISE_IS_NON_STD Code: PA 4991

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4992-- {text}: Use of concatenation operator is non-standard.

Condition Identifier: PAE\$CONCAT_OP_USAGE_NON_STD Code: PA 4992

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary. No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4993-- {text}: Comparison of STRING and CHAR is non-standard.

Condition Identifier: PAE\$STR_CHAR_COMPARE_IS_NON_STD Code: PA 4993

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4994-- {text}: Comparison of unequal size strings is non-standard.

Condition Identifier: PAE\$UNEQUAL_STR_COMPARE_NON_STD Code: PA 4994

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4995-- {text}: Assignment of STRING and CHAR is non-standard.

Condition Identifier: PAE\$STR_CHAR_ASSIGN_NON_STD Code: PA 4995

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4996-- {text}: Assignment of unequal size strings is non-standard.

Condition Identifier: PAE\$UNEQUAL_STR_SIZE_NON_STD Code: PA 4996

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4997-- {text}: Duplication of declaration section is non-standard.

Condition Identifier: PAE\$DUP_DECL_SECTION_IS_NON_STD Code: PA 4997

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4998-- {text}: Relaxed order of declarations is non-standard.

Condition Identifier: PAE\$DECL_RELEASED_ORDER_NON_STD Code: PA 4998

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--FATAL PA 4999-- {text}: The VALUE initialization part is non-standard.

Condition Identifier: PAE\$VALUE_DECL_PART_NON_STD Code: PA 4999

Description: The indicated feature does not follow ISO or ANSI/IEEE standards.

User Action: No action is necessary for NOS/VE Pascal. For portability, this feature should not be used.

--INFORMATIVE PA 5000-- No error. This message should never be printed.

Condition Identifier: PAE\$NO_ERROR Code: PA 5000

Description: This is not an error.

User Action: No action is necessary.

--FATAL PA 5006-- The value 0 is not allowed as the right operand to divide.

Condition Identifier: PAE\$DIVIDE_FAULT Code: PA 5006

Description: The problem was caused by an attempt to divide a number by zero. This is illegal.

User Action: Make sure the arithmetic operations do not involve division by zero.

--FATAL PA 5007-- Attempted DISPOSE of a pointer variable with an invalid value: {text}.

Condition Identifier: PAE\$DISPOSE_INVALID_VALUE Code: PA 5007

Description: A pointer variable with an invalid value cannot be disposed of. A pointer must be initialized with the NEW command first.

User Action: Check program logic.

--FATAL PA 5008-- Attempted DISPOSE of a NIL pointer.

Condition Identifier: PAE\$DISPOSE_NIL_VALUE Code: PA 5008

Description: A NIL pointer cannot be disposed of because it points to no location in memory.

User Action: Check program logic.

--FATAL PA 5011-- Internal Error: Cannot allocate PDT on stack.

Condition Identifier: PAE\$CANT_ALLOCATE_PDT Code: PA 5011

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5012-- Internal Error: Cannot allocate file table.

Condition Identifier: PAE\$CANT_ALLOC_FILE_TABLE Code: PA 5012

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5013-- Internal Error: Cannot allocate wsa in REWRITE for file {text}.

Condition Identifier: PAE\$REWRITE_WSA_ALLOC_ERR Code: PA 5013

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5014-- Internal Error: Cannot allocate wsa in RESET for file {text}.

Condition Identifier: PAE\$RESET_WSA_ALLOC_ERR Code: PA 5014

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5018-- EOF must be TRUE before PUT on file {text}.

Condition Identifier: PAE\$PUT_WITHOUT_EOF Code: PA 5018

Description: A REWRITE command must be done first before CALLING the PUT procedure.

User Action: Check program action.

--FATAL PA 5020-- EOF must be true prior to use of PAGE on file {text}.

Condition Identifier: PAE\$PAGE_WITHOUT_EOF Code: PA 5020

Description: A REWRITE command has to be done first before calling the PAGE procedure.

User Action: Check program logic.

--FATAL PA 5022-- EOF must be true prior to use of WRITE or WRITELN on file {text}.

Condition Identifier: PAE\$WRITE_WITHOUT_EOF Code: PA 5022

Description: A REWRITE command has to be done first before executing the WRITE or WRITELN command.

User Action: Check program logic. Do a REWRITE command before executing the WRITE or WRITELN command.

--FATAL PA 5024-- EOLN activated when EOF is TRUE for file {text} at line {text}.

Condition Identifier: PAE\$EOLN_WHEN_EOF Code: PA 5024

Description: EOLN has already been inspected. Must not attempt to test EOLN again.

User Action: Check program logic.

--FATAL PA 5026-- Input of REAL number {text} from file {text} yields {text} error.

Condition Identifier: PAE\$INPUT_FLOATING_ERROR Code: PA 5026

Description: The REAL number exponent is either too large or too small.

User Action: Check the REAL number input. Change the exponent value so that is within the required range.

--FATAL PA 5027-- File {text} is undefined prior to use of GET.

Condition Identifier: PAE\$GET_ON_UNDEF_FILE Code: PA 5027

Description: The file{text} must be created first before calling the GET procedure.

User Action: Be sure to create the file{text} first.

--FATAL PA 5028-- File {text} is undefined prior to use of PAGE.

Condition Identifier: PAE\$PAGE_ON_UNDEF_FILE Code: PA 5028

Description: The file{text} must be created first before calling the PAGE procedure.

User Action: Be sure to create the file{text} first.

--FATAL PA 5029-- File {text} is undefined prior to use of PUT.

Condition Identifier: PAE\$PUT_ON_UNDEF_FILE Code: PA 5029

Description: The file{text} must be created first before calling the PUT procedure.

User Action: Be sure to create the file{text} first.

--FATAL PA 5030-- File {text} is undefined prior to use of READ.

Condition Identifier: PAE\$READ_ON_UNDEF_FILE Code: PA 5030

Description: The file{text} must be created first before executing the READ command.

User Action: Be sure to create the file{text} first.

--FATAL PA 5031-- File {text} is undefined prior to use of WRITE or WRITELN.

Condition Identifier: PAE\$WRITE_ON_UNDEF_FILE Code: PA 5031

Description: The file{text} must be created first before executing the WRITE command.

User Action: Be sure to create the file{text} first.

--FATAL PA 5033-- File mode must be GENERATION prior to use of PAGE on file {text}.

Condition Identifier: PAE\$PAGE_WITH_WRONG_MODE Code: PA 5033

Description: A REWRITE command must be done first before calling the PAGE procedure.

User Action: Do a REWRITE command before calling the procedure.

--FATAL PA 5034-- File mode must be GENERATION prior to use of PUT on file {text}.

Condition Identifier: PA\$PUT_WITH_WRONG_MODE Code: PA 5034

Description: A REWRITE command must be done before calling the PUT procedure.

User Action: Do a REWRITE command before calling the procedure.

--FATAL PA 5035-- File mode must be GENERATION prior to use of WRITE or WRITELN on file {text}.

Condition Identifier: PA\$WRITE_WITH_WRONG_MODE Code: PA 5035

Description: A REWRITE command must be done before executing the WRITE command.

User Action: Do a REWRITE command before executing the WRITE command.

--FATAL PA 5037-- File mode must be INSPECTION prior to use of GET on file {text}.

Condition Identifier: PA\$GET_WITH_WRONG_MODE Code: PA 5037

Description: A RESET command must be done before calling the GET procedure.

User Action: Do a RESET command before calling the procedure.

--FATAL PA 5038-- File mode must be INSPECTION prior to use of READ on file {text}.

Condition Identifier: PA\$READ_WITH_WRONG_MODE Code: PA 5038

Description: A RESET command must be done before executing the READ command.

User Action: Do a RESET command before executing the READ command.

--FATAL PA 5039-- The Fractional Digits value must be greater than or equal to 1. The value {text} is not allowed.

Condition Identifier: PA\$FRAC_DIGITS_LT_ONE Code: PA 5039

Description: The value of the Fractional Digit must be greater than or equal to 1.

User Action: Check program logic. Fractional digit must be greater than or equal to one.

--FATAL PA 5040-- Free of unallocated block in DISPOSE.

Condition Identifier: PA\$FREE_OF_UNALLOC_BLOCK Code: PA 5040

Description: Tried to dispose of a block that no longer existed.

User Action: Check program logic.

--FATAL PA 5041-- GET attempted on file {text} when EOF is TRUE.

Condition Identifier: PA\$GET_WHEN_EOF Code: PA 5041

Description: EOF condition must be tested before attempting to call the GET procedure.

User Action: Test the EOF condition first, then call the GET procedure. Read the section in the Pascal manual on the GET procedure.

--FATAL PA 5044-- Input attempted after end of file {text} using GET.

Condition Identifier: PA\$GET_AFTER_EOF Code: PA 5044

Description: Input must not be entered after EOF has been reached.

User Action: Check program logic.

--FATAL PA 5046-- Internal Error: insufficient space to perform textfile output on file {text}.

Condition Identifier: PA\$WSA_OVERFLOW_ON_OUTPUT Code: PA 5046

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5047-- The INTEGER value {text} read from file {text} exceeds MAXINT.

Condition Identifier: PAE\$INTEGER_EXCEEDS_MAXINT Code: PA 5047

Description: The INTEGER value must not exceed the declared maximum integer value.

User Action: Reduce the INTEGER value so it does not exceed the declared maximum integer value.

--FATAL PA 5049-- Interactive file {text} must be a textfile.

Condition Identifier: PAE\$TERMINAL_FILE_NOT_TEXTFILE Code: PA 5049

Description: The interactive file is not a textfile. The only type of interactive file is a textfile.

User Action: Correct interactive file so it is a textfile. Check program logic.

--FATAL PA 5054-- The value '{text}' read from file {text} is not a valid REAL number.

Condition Identifier: PAE\$INVALID_REAL_NUMBER Code: PA 5054

Description: The value that was read from the file is not a valid REAL number.

User Action: Make sure the value is a valid REAL number. Read the chapter on valid REAL numbers.

--FATAL PA 5055-- Line length must be <= 150 for interactive input on file {text}.

Condition Identifier: PAE\$LINE_LEN_GT_150 Code: PA 5055

Description: The line length is too long for interactive input. The length must not be greater than 150.

User Action: Shorten the line length so it is less than 150.

--FATAL PA 5056-- Internal Error: Lower merge error in DISPOSE.

Condition Identifier: PAE\$DISPOSE_LOWER_MERGE Code: PA 5056

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5057-- More than 100 files used.

Condition Identifier: PAE\$TOO_MANY_FILES Code: PA 5057

Description: The maximum number of files that can be used is 100, due to current system limitations.

User Action: Reduce the number of files.

--FATAL PA 5061-- READ attempted when EOF is TRUE on file {text}.

Condition Identifier: PAE\$READ_AFTER_EOF Code: PA 5061

Description: Failure to do a reset or there was insufficient data for the program.

User Action: Check program logic. Try putting in a "while not eof" statement.

--FATAL PA 5062-- RESET attempted on undefined file {text}.

Condition Identifier: PAE\$RESET_ON_UNDEF_FILE Code: PA 5062

Description: A file{text} must exist before a RESET command can be executed.

User Action: Create the file before attempting a RESET command.

--FATAL PA 5063-- The value {text} is not allowed as the right operand to MOD at line {text}. The value must be greater than zero.

Condition Identifier: PAE\$MOD_ARG_LT_ONE Code: PA 5063

Description: The selected value is not a valid right operand. The value cannot be less than or equal to zero.

User Action: Check program logic. Make sure the value is greater than zero. Refer to the section in the Pascal manual that discusses operands.

--FATAL PA 5067-- The value '{text}' read from file {text} is not a valid signed number.

Condition Identifier: PAE\$INVALID_SIGNED_NUMBER Code: PA 5067

Description: The value in the file is not a valid signed number.

User Action: Make sure the value is a valid signed number. Read the section in the Pascal manual that discusses signed numbers.

--FATAL PA 5068-- The TotalWidth value must be greater than or equal to 1. The value {text} is not allowed.

Condition Identifier: PAE\$TOTAL_WIDTH_LT_ONE Code: PA 5068

Description: The Total Width value is invalid because the value is less than 1.

User Action: Ensure that the Total Width value is greater than or equal to one.

--FATAL PA 5069-- Internal error: Unable to allocate space for NEW variable in Heap.

Condition Identifier: PAE\$CANT_ALLOCATE_VAR Code: PA 5069

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5070-- Internal error: Upper merge error in DISPOSE.

Condition Identifier: PAE\$DISPOSE_UPPER_MERGE Code: PA 5070

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5071-- System condition detected: Arithmetic overflow at P register = {text}.

Condition Identifier: PAE\$ARITHMETIC_OVERFLOW Code: PA 5071

Description: The resulting number from the arithmetic operation was too large.

User Action: Reduce the size of your numbers, or try altering the order of the arithmetic operations (don't combine two large numbers), or do self-scaling.

--FATAL PA 5073-- System condition detected: Exponent overflow at P register = {text}.

Condition Identifier: PAE\$EXPONENT_OVERFLOW Code: PA 5073

Description: The exponent value was too large.

User Action: Use an exponent that is within the declared limit.

--FATAL PA 5074-- System condition detected: Exponent underflow at P register = {text}.

Condition Identifier: PAE\$EXPONENT_UNDERFLOW Code: PA 5074

Description: The exponent value was too small (probably too close to zero).

User Action: Use an exponent that is within the declared limit.

--FATAL PA 5075-- System condition detected: Floating point indefinite at P register = {text}.

Condition Identifier: PAE\$FP_INDEFINITE Code: PA 5075

Description: There was an attempt to use an uninitialized variable. This is illegal.

User Action: Check the variable to find out if it was initialized to a value before it was used.

--FATAL PA 5076-- System condition detected: Invalid BDP data at P register = {text}.

Condition Identifier: PAE\$INVALID_BDP_DATA Code: PA 5076

Description: The BDP (Business Data Processing) data cannot be passed to a Pascal routine from a Cobol routine.

User Action: Check the external routine that was called and used the BDP instructions.

--FATAL PA 5077-- Attempted output of REAL number to file {text} yields {text} error.

Condition Identifier: PAE\$OUTPUT_FLOATING_NUMBER_ERR Code: PA 5077

Description: Invalid result when sent to math library conversion routine.

User Action: Check the numeric value.

--FATAL PA 5078-- Attempted output of INTEGER number to file {text} yields {text} error.

Condition Identifier: PAE\$OUTPUT_INTEGER_ERR Code: PA 5078

Description: Invalid result when sent to math library conversion routine.

User Action: Check the numeric value.

--FATAL PA 5079-- Size of binary input item must be $\leq 2^{31} - 1$ bytes, current size is {text}.

Condition Identifier: PAE\$BINARY_INPUT_TOO_LARGE Code: PA 5079

Description: The size of the binary input item is too large. The size cannot exceed $2^{31} - 1$ bytes.

User Action: Reduce the size of the binary input item.

--FATAL PA 5080-- Internal Error: Error in math library routine MLP\$MOVE_BYTES: {text}.

Condition Identifier: PAE\$MOVE_BYTES_ERROR Code: PA 5080

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5081-- RESET on file {text} failed because EOF occurred before buffer variable filled.

Condition Identifier: PAE\$RESET_AFTER_EOF Code: PA 5081

Description: A RESET command is not allowed when EOF is true.

User Action: Check data file.

--FATAL PA 5082-- Size of binary output item must be $\leq 2^{31} - 1$ bytes, current size is {text}.

Condition Identifier: PAE\$BINARY_OUTPUT_TOO_LARGE Code: PA 5082

Description: The size of the binary output item is too large. The size cannot exceed $2^{31} - 1$ bytes.

User Action: Reduce the size of the binary output item.

--FATAL PA 5085-- Program terminated by calling HALT.

Condition Identifier: PAE\$PROGRAM_HALTED Code: PA 5085

Description: Executing the HALT command terminated the program that was being executed.

User Action: If you did not want the program to terminate, do not call HALT. Run the program again.

--FATAL PA 5086-- Internal Error: Cannot find stack frame for target label of GOTO statement.

Condition Identifier: PAE\$GOTO_TARGET_NOT_FOUND Code: PA 5086

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5087-- File buffer variable is undefined for file {text}.

Condition Identifier: PA\$FILE_BUFFER_UNDEFINED Code: PA 5087

Description: File buffer variable must be defined for the file{text}. File was mispositioned.

User Action: Check program flow or program environment.

--FATAL PA 5089-- Input of integer value {text} from file {text} yields {text} error.

Condition Identifier: PA\$INPUT_INTEGER_ERR Code: PA 5089

Description: Integer entered was too large. The number must be smaller.

User Action: Check the integer input. Change the integer so that it is a smaller value.

--FATAL PA 5090-- Input line length {text} cannot be greater than page width of {text} for interactive file {text}.

Condition Identifier: PA\$LINE_LEN_GT_PAGE_WIDTH Code: PA 5090

Description: The line length is too long. The length must be shortened so it is not any wider than the page width.

User Action: The length of the input line should be shortened to no wider than the page width.

--FATAL PA 5091-- Character value '{text}' read from file {text} is outside the declared subrange of {text} .. {text}.

Condition Identifier: PA\$INPUT_CHAR_RANGE_ERR Code: PA 5091

Description: The declared subrange does not include the indicated value.

User Action: Change the character value so that it is within the declared subrange of [text]..[text].

--FATAL PA 5092-- Integer value {text} read from file {text} is outside the declared range of {text} .. {text}.

Condition Identifier: PA\$INPUT_INTEGER_RANGE_ERR Code: PA 5092

Description: The integer value[text] read from the file[text] must not be outside the declared range of [text]..[text].

User Action: Ensure that the integer value is within the declared range of [text]..[text].

--FATAL PA 5093-- The value {text} is out of range at line {text}.

Condition Identifier: PA\$INTEGER_RANGE_ERROR Code: PA 5093

Description: The integer value must be within the specified range.

User Action: Ensure that the integer value is within the specified range.

--FATAL PA 5094-- The ordinal value {text} is out of range at line {text}.

Condition Identifier: PA\$ORDINAL_RANGE_ERROR Code: PA 5094

Description: The ordinal value must be within the specified range.

User Action: Ensure that the ordinal value is within the specified range.

--FATAL PA 5095-- The character value '{text}' is out of range at line {text}.

Condition Identifier: PA\$CHAR_RANGE_ERROR Code: PA 5095

Description: The character value must be within the specified range.

User Action: Change the character value so that it falls within the specified range.

--FATAL PA 5096-- The subscript value {text} is out of range at line {text}.

Condition Identifier: PA\$INTEGER_SUBSCRIPT_ERROR Code: PA 5096

Description: The subscript value must be within the specified range.

User Action: Ensure that the subscript value is within the specified range.

--FATAL PA 5097-- The ordinal subscript value {text} is out of range at line {text}.

Condition Identifier: PA\$ORDINAL_SUBSCRIPT_ERROR Code: PA 5097

Description: The ordinal subscript value must be within the specified range.

User Action: Ensure that the ordinal subscript value is within the specified range.

--FATAL PA 5098-- The character subscript value '{text}' is out of range at line {text}.

Condition Identifier: PA\$CHAR_SUBSCRIPT_ERROR Code: PA 5098

Description: The character subscript value is not within the required range.

User Action: Check program logic. Change the character subscript value [text] so that it is within the required range.

--FATAL PA 5099-- Attempted dereference of a pointer with an invalid value: {text} at line {text}.

Condition Identifier: PA\$DEREF_INVALID_POINTER Code: PA 5099

Description: Must initialize a pointer with NEW command first before dereferencing. A pointer also cannot be dereferenced if the value it was pointing to was already disposed of.

User Action: Check logic pertaining to the creation of the variable.

--FATAL PA 5100-- Attempted dereference of a NIL pointer at line {text}.

Condition Identifier: PA\$DEREF_NIL_POINTER Code: PA 5100

Description: A NIL pointer cannot be dereferenced because it points at no location in memory.

User Action: Do not dereference a nil pointer. Check program logic.

--FATAL PA 5101-- The ordinal value of CHAR types must be within the range of 0..255. The ordinal value {text} is outside this range at line {text}.

Condition Identifier: PA\$INVALID_CHAR Code: PA 5101

Description: The ordinal value is not within the required range 0..255.

User Action: Change the ordinal value so that it is within the range 0..255.

--FATAL PA 5102-- The ordinal value of CHAR types must be within the range of 0..255. The character subscript with ordinal value {text} is outside this range at line {text}.

Condition Identifier: PA\$INVALID_CHAR_SUBSCRIPT Code: PA 5102

Description: The ordinal value is not within the required range of 0..255.

User Action: Change the ordinal value (char. subscript) so that it is within the range 0..255.

--FATAL PA 5103-- The CASE selector value {text} does not match any of the CASE constants ending at line {text}.

Condition Identifier: PA\$INT_CASE_SELECTOR_ERR Code: PA 5103

Description: The CASE selector value does not match any of the CASE constants in the CASE statement.

User Action: Check program logic. Change the integer CASE selector value so that it matches one of the CASE constants.

--FATAL PA 5104-- The character case selector value '{text}' does not match any of the case constants at line {text}.

Condition Identifier: PA\$CHAR_CASE_SELECTOR_ERR Code: PA 5104

Description: The character case selector value does not match any of the CASE constants in the CASE statement.

User Action: Check program logic. The character CASE selector value must match one of the CASE constants.

--FATAL PA 5105-- The source string size of {text} is greater than the target string size at line {text}.

Condition Identifier: PA\$SOURCE_STRING_GT_TARGET Code: PA 5105

Description: The source string size must not be greater than the target string size. The problem could involve incorrect parameter passing.

User Action: Ensure that the source string size is not greater than the target size. Check the passing of parameters.

--FATAL PA 5106-- The value {text} is not acceptable as a substring index at line {text}. Substring index must be ≥ 1 .

Condition Identifier: PA\$SUBSTRING_LOWBOUND_ERROR Code: PA 5106

Description: The substring index must be greater than or equal to one.

User Action: Ensure that the substring index is greater than or equal to one.

--FATAL PA 5107-- Lower substring index {text} must be \leq upper substring index at line {text}.

Condition Identifier: PA\$SUBSTRING_RANGE_ERROR Code: PA 5107

Description: The lower substring index cannot be greater than the upper substring index.

User Action: Change the lower substring index so it is less than or equal to the upper string index.

--FATAL PA 5108-- Upper substring index {text} must be \leq length of the string at line {text}.

Condition Identifier: PA\$SUBSTRING_UPPERBOUND_ERROR Code: PA 5108

Description: The length of the string must be greater than or equal to the upper substring index.

User Action: Ensure that the string length is greater than or equal to the upper substring index.

--FATAL PA 5109-- Length of concatenated string is {text} at line {text}. Maximum string length allowed is 65535.

Condition Identifier: PA\$MAX_STRING_LEN_OVERFLOW Code: PA 5109

Description: The maximum string length permitted is 65535. The length of concatenated string is too long.

User Action: Shorten the length of the concatenated string.

--FATAL PA 5110-- The value {text} (substring start position plus substring length - 1) exceeds the length of the string expression at line {text}.

Condition Identifier: PA\$SUBSTR_LEN_ERROR Code: PA 5110

Description: Self-explanatory.

User Action: Check the value length and correct it so that it is not longer than the length of the string expression.

--FATAL PA 5111-- Substring length of {text} is incorrect at line {text}. Substring length cannot be less than zero.

Condition Identifier: PA\$SUBSTR_LEN_NEGATIVE Code: PA 5111

Description: Substring length of {text} must not be less than zero.

User Action: Change substring length to a number greater than zero.

--FATAL PA 5112-- The value {text} is out of range at line {text}. The length of a dynamic string must be in the range of 1 .. 65535.

Condition Identifier: PA\$DYNAMIC_STRING_RANGE_ERROR Code: PA 5112

Description: The dynamic string length must be within the range of 1..65535.

User Action: Ensure that the dynamic string length is within the range of 1..65535.

--FATAL PA 5113-- Value for total width: {text} must be greater than or equal to zero for STRING types.

Condition Identifier: PAE\$TOTAL_WIDTH_LT_ZERO Code: PA 5113

Description: The Total width value is invalid because the value is less than zero. For STRING types, this value must be greater than or equal to zero.

User Action: Ensure that the total width value is greater than or equal to zero for STRING types.

--FATAL PA 5114-- Direct file {text} is empty. A GET cannot be performed on an empty file.

Condition Identifier: PAE\$GET_ON_EMPTY_FILE Code: PA 5114

Description: The direct file does not contain any data. The GET command cannot transfer data if there is no RESET call or if there is no data.

User Action: Make sure the file contains data before executing a GET command.

--FATAL PA 5115-- Direct file {text} is empty. A READ cannot be performed on an empty file.

Condition Identifier: PAE\$READ_ON_EMPTY_FILE Code: PA 5115

Description: The direct file does not contain any data. The READ command cannot assign any values unless the file contains some type of data.

User Action: Make sure the file contains data before executing a READ command.

--FATAL PA 5116-- Sequential file {text} must be at EOF to allow a PUT.

Condition Identifier: PAE\$NEED_EOF_FOR_SEQ_PUT Code: PA 5116

Description: In order to execute a PUT operation on a sequential file, it must be at EOF (end-of-file).

User Action: Check for EOF (end-of-file) on the sequential file first, then execute the PUT operation on the file.

--FATAL PA 5117-- Sequential file {text} must be at EOF to allow a WRITE.

Condition Identifier: PAE\$NEED_EOF_FOR_SEQ_WRITE Code: PA 5117

Description: In order to execute a WRITE operation on a sequential file, it must be at EOF (end-of-file).

User Action: Check for EOF (end-of-file) on the sequential file first, then execute the WRITE operation on the file.

--FATAL PA 5118-- Direct file {text} is positioned beyond upper index. No data can be written to the file at this position.

Condition Identifier: PAE\$PUT_BEYOND_EOF Code: PA 5118

Description: When calling a direct file from another program, do not position the file beyond the declared upper index. It is impossible to write any data to the file.

User Action: Check the index range of the file that is being called from another program. Make sure it is within that declared range. For example:

```
Prog 1:  VAR direct_file1 : FILE [1..50] OF integer;  
Prog 2:  write (direct_file1,70)
```

The upper index of 70 is too large.

--FATAL PA 5119-- Length of existing direct file {text} exceeds the length determined by the index range specified.

Condition Identifier: PA\$LENGTH_EXCEEDS_RANGE Code: PA 5119

Description: When calling a direct file from another program, do not use an index range larger than the declared range.

User Action: Check the index range of the file that is being called from the other program. Make sure it is within the declared range.

For example:

```

Prog1: VAR file_1 : FILE [1..10] OF integer;
Prog2: read (file_1,12);

```

The index of 12 exceeds the range of 1..10.

--FATAL PA 5120-- Internal error: File {text} is not empty and the buffer size is 0.

Condition Identifier: PA\$ZERO_BUFFER_SIZE Code: PA 5120

Description: This is an internal error.

User Action: Follow site defined procedure for reporting internal software errors.

--FATAL PA 5121-- Index value given is not within index range defined for direct file {text}.

Condition Identifier: PA\$INDEX_OUT_OF_RANGE Code: PA 5121

Description: When calling a direct file from another program, do not use an index value that has not been defined for that file.

User Action: Check the index value of the program being called from another program. Make sure it falls within the range of the defined file.

Example:

```

Prog1: VAR file_name : FILE [1..10] OF integer;
Prog2: read (file_name,-4);

```

The value is not within the range of 1..10.

--FATAL PA 5122-- Direct file {text} is empty. LASTPOSITION for an empty file is undefined.

Condition Identifier: PA\$LASTPOSITION_ON_EMPTY_FILE Code: PA 5122

Description: The direct file contains no data. You cannot execute the LASTPOSITION command if the file is empty because there is no value to return.

User Action: Check to make sure the file contains data before executing a LASTPOSITION command.

--FATAL PA 5123-- Cannot position direct file {text} beyond current end of information. No gaps in information are allowed in a file.

Condition Identifier: PA\$POSITION_BEYOND_EOI Code: PA 5123

Description: When appending a file to another file, you have to append it directly onto the end of the first file, with no gaps in between the two files.

User Action: Check the positioning of the file being appended. Be sure to append directly onto the end of the original file, leaving no gaps.

--FATAL PA 5124-- Direct file {text} is empty. POSITION for an empty file is undefined.

Condition Identifier: PAE\$POSITION_ON_EMPTY_FILE Code: PA 5124

Description: The direct file contains no data. You cannot execute the POSITION command if the file is empty because there is no value to position at.

User Action: Check to make sure the file contains data before executing a POSITION command. Then execute POSITION command to set the file up for inquiry.

--FATAL PA 5125-- File {text} must be of type direct access for seek operation.

Condition Identifier: PAE\$SEEK_ON_DIRECT_ONLY Code: PA 5125

Description: In order to perform a seek type of operation (seekread, seekwrite, seekupdate), the file must be of type direct access, not sequential.

User Action: Determine what type of file is appropriate (sequential or direct access) before attempting a seek operation.

--FATAL PA 5126-- Direct file {text} is undefined. SEEKREAD cannot be performed on an undefined file.

Condition Identifier: PAE\$SEEKREAD_ON_UNDEF_FILE Code: PA 5126

Description: The operation SEEKREAD cannot be performed on a file if it has not been opened.

User Action: Open the direct file first by doing a REWRITE or SEEKWRITE to open the file. Then execute a SEEKREAD operation.

--FATAL PA 5127-- Direct file {text} is undefined. POSITION cannot be performed on an undefined file.

Condition Identifier: PAE\$POSITION_ON_UNDEF_FILE Code: PA 5127

Description: The operation POSITION cannot be performed on a file if it has not been opened. Open the file first with a REWRITE or a SEEKWRITE operation, then execute a POSITION operation.

User Action: Open the direct file first with a rewrite or seekwrite, then execute a POSITION operation.

--FATAL PA 5128-- Direct file {text} is undefined. LASTPOSITION cannot be performed on an undefined file.

Condition Identifier: PAE\$LASTPOS_ON_UNDEF_FILE Code: PA 5128

Description: The operation LASTPOSITION cannot be performed on a file if it has not been opened.

User Action: Open the direct file first with a rewrite, seekwrite, or seekupdate, then execute a LASTPOSITION operation.

--FATAL PA 5129-- Direct file {text} is undefined. SEEKWRITE or SEEKUPDATE operation will create this file, but the position given, {text}, is not equal to the index lower bound.

Condition Identifier: PAE\$INDEX_NOT_LOWER_BOUND Code: PA 5129

Description: When calling a direct file from another program, after opening the file, make sure that the position of the file is within the declared range.

User Action: Check the index range of the file that is being called from another program.

Example:

```
Prog1: VAR file_1 : FILE [1..10] OF integer;  
Prog2: read (file_1,-3);
```

The index value -3 is not the lower index bound.

--FATAL PA 5130-- SEEKREAD on the empty file, {text}, must give the index lower bound for the index value.

Condition Identifier: PA\$READ_INDEX_LOWER_BOUND Code: PA 5130

Description: When executing a SEEKREAD operation, the index used was not the lower bound for the index range. This is illegal.

User Action: Make sure the index used is the lower bound of the index range, if the file is empty. So if the file is empty, this allows you to continue processing other files. Remember, this index must be the lower bound of the index range.

Example:

```

PROGRAM test
  VAR f : FILE [1..10] OF integer;
  .
  .
  seekread (f,i)    {i must be 1 or the lower bound}
  .
  .
  read (f,var_name);
  .
  .

```

--FATAL PA 5131-- File {text} must be defined for the function {text}.

Condition Identifier: PA\$UNDEFINED_FILE Code: PA 5131

Description: Open the file first before having a function call it.

User Action: Use a REWRITE command to open the file first, then the function can call it.

--FATAL PA 5132-- Value {text} from POSITION is out of range at line {text}.

Condition Identifier: PA\$POSITION_VAL_RANGE_ERROR Code: PA 5132

Description: The value returned from the POSITION function is not within the valid range.

User Action: Check the range defined for the POSITION function. Use a value that is within that valid range.

--FATAL PA 5133-- Attempt made to bind file {text} while a binding exists.

Condition Identifier: PA\$FILE_ALREADY_BOUND Code: PA 5133

Description: When a binding already exists, it is illegal to attempt to bind that file {text} again.

User Action: To bind the file {text}, you must unbind the file first, then bind the file correctly.

--FATAL PA 5134-- Attempt made to apply {text} to file {text} when file is not bound.

Condition Identifier: PA\$OPERATION_ON_UNBOUND_FILE Code: PA 5134

Description: An attempt was made to apply text to a file when the file has not been bound (or connected) to the file reference.

User Action: First, bind the file to the Pascal file reference, then apply {text} to the file.

--FATAL PA 5135-- Attempt to use component of an inactive variant. Tag ordinal value is {text} at line {text}.

Condition Identifier: PA\$VARIANT_INACTIVE Code: PA 5135

Description: It is invalid to use a component of a variant unless the variant is active and defined.

User Action: Define the variant before attempting to use the component.

--FATAL PA 5136-- The size of the array passed to PAP\$DEFINE_APPLIC_UNIT_ARRAY must be from 1 to 63 full words.

Condition Identifier: PAE\$APPLIC_ARRAY Code: PA 5136

Description: The array being used as a parameter for the application unit array procedure must be defined with the type ARRAY [1..63] OF integer.

User Action: Correct the size of the array.

--ERROR PE 315-- Terminal model not specified (SET_TERMINAL_ATTRIBUTES, SETTA) or the terminal definition for the model specified is not in your library list.

Condition Identifier: PEE\$NO_SCREEN_LEVEL Code: PE 315

Description: You have not specified a terminal model or the terminal definition is not in the terminal definition directory.

User Action: Set your terminal model using the Set_Terminal_Attributes command.

--INFORMATIVE PE 330-- Internal error: INVALID SCREEN TRANSITION. Please bring this to the attention of your site analyst.

Condition Identifier: PEE\$INVALID_SCREEN_TRANSITION Code: PE 330

Description: There is a software problem in the Programming Environment or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 765-- Loader errors were found, however a failure while attempting to weave them with the source program prevents the Programming Environment from displaying them. Try finding them on the LOADMAP component.

Condition Identifier: PEE\$WEAVE_LOADER_ERROR_FAILURE Code: PE 765

Description: There is a software problem in the Programming Environment or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 775-- Compilation errors were found, however a failure while attempting to weave them with the source program prevents the Programming Environment from displaying them. Try finding them on the COMPILATION_LISTING component.

Condition Identifier: PEE\$WEAVE_FAILED Code: PE 775

Description: There is a software problem in the Programming Environment or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 900-- Programming Environment Internal Error -- The Systems Program Description for the Programming Environment has been altered. It must contain no object file list or module list. It must specify a library list where the first library contains the bound product, program descriptions, and SCL procedures, and the second library contains SCL procedures used with Edit_File.

Condition Identifier: PEE\$ENTPE_PROGRAM_DESC_ERROR Code: PE 900

Description: There is a software problem in the Programming Environment or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 905-- Programming Environment Internal Error -- No room on CYBIL run-time stack to hold the enter_programming_environment program description.

Condition Identifier: PEE\$NO_ROOM_FOR_PROG_DESC Code: PE 905

Description: There is a software problem in the Programming Environment product or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 910-- Programming Environment Internal Error -- Invalid post editor action.

Condition Identifier: PEE\$INVALID_POST_EDIT_ACTION Code: PE 910

Description: There is a software problem in the Programming Environment or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--CATASTROPHIC PE 912-- Programming Environment Internal Error -- System heap ran out of room while obtaining editor menu list.

Condition Identifier: PEE\$NO_ROOM_FOR_EDIF_MENU_LIST Code: PE 912

Description: There is a software problem in the Programming Environment product or the product has been improperly installed.

User Action: Talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--INFORMATIVE PE 1700-- Programming Environment terminated.

Condition Identifier: PEE\$INTERRUPT_TERMINATE Code: PE 1700

Description: A user_break_2 condition (terminate_break) was received by the Programming Environment. This caused the Programming Environment to abort what it was doing, clean up, and terminate.

User Action: None. This is an informative message. You entered a key sequence that was interpreted by the network as an interrupt and communicated to NOS/VE as a user_break_2 condition.

--ERROR PE 1710-- The application_menu {text} contains only menu classes it must also contain menu items. Check your application menus in your command list.

Condition Identifier: PEE\$MENU_CLASS_LIST_BOTH_NIL Code: PE 1710

Description: The indicated application menu resides in a message module on your command list. You have either incorrectly created it or the Programming Environment has been improperly installed.

User Action: Use the menu list in the message module PEM\$US_ENGLISH in \$\$SYSTEM.OSF\$COMMAND_LIBRARY as an example of how to correct your application menu. Or, talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--ERROR PE 1720-- The application_menu {text} contains {text} menu items, it must contain exactly, {text} menu items. Check your application menus in your command list.

Condition Identifier: PEE\$MENU_LIST_WRONG_SIZE Code: PE 1720

Description: The indicated application menu resides in a message module on your command list. You have either incorrectly created it or the Programming Environment has been improperly installed.

User Action: Use the menu list in the message module PEM\$US_ENGLISH in \$\$SYSTEM.OSF\$COMMAND_LIBRARY as an example of how to correct your application menu. Or, talk to your site personnel about the problem and, if appropriate, send a PSR to Control Data. PE8 is the Programming Environment product identifier.

--ERROR PE 1725-- The compilation parameters have been lost for this program. Use the Done function of the CPARAM screen to reestablish them.

Condition Identifier: PEE\$RECREATE_\$COMPILE_PARMS Code: PE 1725

Description: The compilation parameters have been lost for this program.

User Action: Use the Done function of the CPARAM screen to reestablish them.

--WARNING PE 1730-- The compilation parameters have been lost for this program. The parameters shown are the compilers defaults. Use the Done function to save the parameter settings for use when compiling. Use the Lists function to retrieve a predefined parameter list such as debug compilation settings or a user defined parameter list.

Condition Identifier: PEE\$COMPILE_PARMS_LOST Code: PE 1730

Description: The compilation parameters have been lost for this program. The parameters shown are the compiler defaults.

User Action: Use the Done function to save the parameter settings for use when compiling. Use the Lists function to retrieve a predefined parameter list such as debug compilation settings or a user defined parameter list.

--ERROR PF 50-- File {file} cannot be attached because it has overflowed to different allocation sizes. The file must be deleted and restored.

Condition Identifier: PFE\$ALLOCATION_MISMATCH Code: PF 50

Description: An attempt was made to attach a file cycle that was created and overflowed to one or more other disk volumes in a previous level of the system. The attachment cannot be completed because the overflow has caused different parts of the file cycle to be written using different allocation sizes.

User Action: Delete the file cycle and restore it.

--ERROR PF 105-- Account name "{text}" is improper.

Condition Identifier: PFE\$BAD_ACCOUNT_NAME Code: PF 105

Description: The account field within the group record contains an invalid name.

User Action: Correct the account field value.

--ERROR PF 111-- Permanent file change type "{text}" is improper.

Condition Identifier: PFE\$BAD_CHANGE_TYPE Code: PF 111

Description: A CHANGE_TYPE field within the change_list array contains an invalid value.

User Action: Correct the CHANGE_TYPE field value.

--ERROR PF 116-- File cycle number "{text}" is out of range. It must be between 1 and 999.

Condition Identifier: PFE\$BAD_CYCLE_NUMBER Code: PF 116

Description: The CYCLE_NUMBER field within the cycle_selector record contains an invalid value. The cycle number must be an integer from 1 through 999.

User Action: Correct the CYCLE_NUMBER field value.

--ERROR PF 122-- File cycle option "{text}" is improper.

Condition Identifier: PFE\$BAD_CYCLE_OPTION Code: PF 122

Description: The valid CYCLE_OPTION field values are PFC\$LOWEST_CYCLE, PFC\$HIGHEST_CYCLE, and PFC\$SPECIFIC_CYCLE.

User Action: Correct the CYCLE_OPTION field value.

--ERROR PF 127-- Family name "{text}" is improper.

Condition Identifier: PFE\$BAD_FAMILY_NAME Code: PF 127

Description: The first name in the path array is invalid. If OSC\$NULL_NAME is specified, the family name of the caller is used.

User Action: Correct the family name.

--ERROR PF 133-- Permanent file group type "{text}" is improper.

Condition Identifier: PFE\$BAD_GROUP_TYPE Code: PF 133

Description: The GROUP_TYPE field in the group record contains an invalid value.

User Action: Correct the GROUP_TYPE field value.

--ERROR PF 138-- Subcatalog name "{text}" is improper.

Condition Identifier: PFE\$BAD_LAST_SUBCATALOG_NAME Code: PF 138

Description: The last name in the path array is invalid.

User Action: Correct the last subcatalog name.

--ERROR PF 144-- Local file name "{text}" is improper.

Condition Identifier: PFE\$BAD_LOCAL_FILE_NAME Code: PF 144

Description: The specified local file name is invalid.

User Action: Correct the local file name with a valid name.

--ERROR PF 150-- Permanent file log option "{text}" is improper.

Condition Identifier: PFE\$BAD_LOG_OPTION Code: PF 150

Description: The valid log identifiers are PFC\$LOG and PFC\$NO_LOG.

User Action: Correct the log value.

--ERROR PF 155-- Master catalog name "{text}" is improper.

Condition Identifier: PFE\$BAD_MASTER_CATALOG_NAME Code: PF 155

Description: The second name in the path array is invalid. If OSC\$NULL_NAME is specified, the master catalog of the caller is used.

User Action: Correct the master catalog name.

--ERROR PF 161-- Subcatalog name "{text}" is improper.

Condition Identifier: PFE\$BAD_NTH_SUBCATALOG_NAME Code: PF 161

Description: The indicated subcatalog name in the path array is invalid.

User Action: Correct the subcatalog name.

--ERROR PF 166-- Permanent file password "{text}" is improper.

Condition Identifier: PFE\$BAD_PASSWORD Code: PF 166

Description: The specified password is an invalid name.

User Action: Correct the password or specify OSC\$NULL_NAME.

--ERROR PF 172-- File name "{text}" is improper.

Condition Identifier: PFE\$BAD_PERMANENT_FILE_NAME Code: PF 172

Description: The specified file name is an invalid name.

User Action: Correct the file name.

--ERROR PF 177-- Project name "{text}" is improper.

Condition Identifier: PFE\$BAD_PROJECT_NAME Code: PF 177

Description: The project field in the group record contains an invalid value.

User Action: Correct the project field value to be a proper name.

--ERROR PF 183-- Permanent file retention period "{text}" is improper.

Condition Identifier: PFE\$BAD_RETENTION_PERIOD Code: PF 183

Description: The retention period value is invalid.

User Action: Correct the retention period value. The retention must be an integer from 1 through 999. The value 999 indicates infinite retention.

--ERROR PF 188-- User name "{text}" is improper.

Condition Identifier: PFES\$BAD_USER_NAME Code: PF 188

Description: The user field of the group record contains an invalid value.

User Action: Correct the user field value.

--ERROR PF 194-- Permanent file wait option "{text}" is improper.

Condition Identifier: PFES\$BAD_WAIT_OPTION Code: PF 194

Description: The valid wait identifiers are PFC\$WAIT and PFC\$NO_WAIT.

User Action: Correct the wait value.

--ERROR PF 207-- Only an administrator may delete a master catalog.

Condition Identifier: PFES\$CANNOT_PURGE_MASTER_CATALOG Code: PF 207

Description: Only an administrator may delete a master catalog.

User Action: Do not attempt to delete a master catalog. The contents of the master catalog, however, may be deleted. Contact the family administrator if it is desired to have the master catalog deleted.

--ERROR PF 214-- Permanent file catalog {text} is full. {text}

Condition Identifier: PFES\$CATALOG_FULL Code: PF 214

Description: No more entries can be defined in the catalog.

User Action: If the full catalog is not the master catalog, define a new subcatalog in the catalog that precedes the full catalog in the catalog path. You can then define new entries in the new subcatalog. Also contact the site analyst.

--ERROR PF 228-- Permanent file catalog {text} is not empty.

Condition Identifier: PFES\$CATALOG_NOT_EMPTY Code: PF 228

Description: The catalog cannot be deleted until all entries in the catalog are deleted.

User Action: Delete each subcatalog and file in the catalog and then delete the catalog.

--ERROR PF 232-- Catalog volume {text} not on-line.

Condition Identifier: PFES\$CATALOG_VOLUME_NOT_ONLINE Code: PF 232

Description: The device on which the catalog resides has been removed from the active configuration.

User Action: Wait for the device to be added back into the active configuration or for the catalog to be restored to an active device.

--ERROR PF 234-- Catalog volume {text} unavailable.

Condition Identifier: PFES\$CATALOG_VOLUME_UNAVAILABLE Code: PF 234

Description: The device on which the catalog resides is temporarily unavailable.

User Action: Wait for the device to become available or for the catalog to be restored to an available device.

--ERROR PF 242-- Cycle number {text} of permanent file {text} is busy. {text}

Condition Identifier: PFES\$CYCLE_BUSY Code: PF 242

Description: The file cycle is currently not available for the access modes or share modes requested.

User Action: Attempt to access the file cycle later or attempt to attach the file with different access modes or share modes.

--ERROR PF 250-- Cycle {text} of file {text} is unavailable because its data resides off-line. Use attach_file to make it available.

Condition Identifier: PFES\$CYCLE_DATA_RESIDES_OFFLINE Code: PF 250

Description: The data for the file cycle resides offline.

User Action: Use the ATTACH_FILE command to explicitly attach the file cycle prior to accessing its data.

--ERROR PF 257-- The highest possible cycle number already exists for file {file}.

Condition Identifier: PFE\$CYCLE_OVERFLOW Code: PF 257

Description: Cycle 999 is already defined for the file.

User Action: Specify another cycle number or delete cycle 999.

--ERROR PF 271-- The lowest possible cycle number already exists for file {file}.

Condition Identifier: PFE\$CYCLE_UNDERFLOW Code: PF 271

Description: Cycle 1 is already defined for the file.

User Action: Specify another cycle number or delete cycle 1.

--ERROR PF 285-- File {text} resides on a medium that is missing. Either delete the cycle, restore it, or wait for the volume to be reinstalled.

Condition Identifier: PFE\$CYCLES_MEDIA_MISSING Code: PF 285

Description: The medium on which the cycle resides is not currently active.

User Action: Either delete the cycle, restore it, or wait for the volume to be reinstalled.

--ERROR PF 350-- Cycle number {text} already exists for permanent file {text}.

Condition Identifier: PFE\$DUPLICATE_CYCLE Code: PF 350

Description: A cycle with the specified cycle number is already defined for the file.

User Action: Specify another cycle number.

--ERROR PF 550-- Family "{text}" already exists.

Condition Identifier: PFE\$FAMILY_ALREADY_EXISTS Code: PF 550

Description: An attempt was made to create a family, but the family name is already defined.

User Action: Ignore the error and use the existing family, or create a different family.

--ERROR PF 825-- Incorrect damage condition. The {text}

Condition Identifier: PFE\$INCORRECT_DAMAGE_CONDITION Code: PF 825

Description: Either a DELETE_DAMAGE_CONDITION of something other than RESPF_MODIFICATION_MISMATCH or PARENT_CATALOG_RESTORED was specified on a call to PFP\$CHANGE, or a DELETE_DAMAGE_CONDITION of RESPF_MODIFICATION_MISMATCH or PARENT_CATALOG_RESTORED was specified on a CHANGE_CATALOG_ENTRY command, or a call to PFP\$CHANGE and the cycle is not in that damage condition.

User Action: Specify DISPLAY_OPTION=CYCLE on a DISPLAY_CATALOG_ENTRY command to determine the damage condition of the cycle. If the cycle has no damage condition, do not try to delete one. If the damage condition is RESPF_MODIFICATION_MISMATCH or PARENT_CATALOG_RESTORED, specify a DELETE_DAMAGE_CONDITION of only RESPF_MODIFICATION_MISMATCH or PARENT_CATALOG_RESTORED on a CHANGE_CATALOG_ENTRY command or a call to PFP\$CHANGE. If the damage condition is not RESPF_MODIFICATION_MISMATCH or PARENT_CATALOG_RESTORED, either restore the cycle or delete it.

--ERROR PF 850-- The password provided is incorrect for permanent file {text}.

Condition Identifier: PFE\$INCORRECT_PASSWORD Code: PF 850

Description: The specified password is not the password defined for the file.

User Action: Ask the file owner to use DISPLAY_CATALOG_ENTRY to determine the correct password.
Correct the password value.

--ERROR PF 875-- The task is executing with insufficient ring privilege for the requested access to permanent file {text}.

Condition Identifier: PFE\$INVALID_RING_ACCESS Code: PF 875

Description: The task is executing with insufficient ring privilege for the required access to the specified permanent file.

User Action: Execute at a lower ring. The ring attributes for the file may be determined with the DISPLAY_FILE_ATTRIBUTES command.

--ERROR PF 1133-- {text} is expected to be the name of a subcatalog, but it is a permanent file.

Condition Identifier: PFE\$LAST_NAME_NOT_SUBCATALOG Code: PF 1133

Description: The last name in the path is the name of a permanent file and not a subcatalog.

User Action: Correct the path.

--ERROR PF 1166-- Local file name "{text}" is already in use.

Condition Identifier: PFE\$LFN_IN_USE Code: PF 1166

Description: The specified local file name is already in use.

User Action: Specify another local file name, or detach the specified file and try again.

--ERROR PF 1314-- {file} is already being used as the name of a file.

Condition Identifier: PFE\$NAME_ALREADY_PERMANENT_FILE Code: PF 1314

Description: The specified name is already being used as the name of a file.

User Action: Specify a different name.

--ERROR PF 1328-- {text} is already being used as the name of a permanent file subcatalog.

Condition Identifier: PFE\$NAME_ALREADY_SUBCATALOG Code: PF 1328

Description: The specified name is already being used as the name of a permanent file subcatalog.

User Action: Specify a different name.

--ERROR PF 1342-- Name "{text}" is already being used.

Condition Identifier: PFE\$NAME_ALREADY_USED Code: PF 1342

Description: The specified name is already being used as the name of either a subcatalog or a file.

User Action: Specify a different name.

--ERROR PF 1357-- {file} is expected to be the name of a file, but it is a catalog.

Condition Identifier: PFE\$NAME_NOT_PERMANENT_FILE Code: PF 1357

Description: The specified name is the name of a subcatalog and not a file.

User Action: Correct the path.

--ERROR PF 1371-- You are not the owner of master catalog {text} of family {text}.

Condition Identifier: PFE\$NOT_MASTER_CATALOG_OWNER Code: PF 1371

Description: Only the owner may create and delete catalogs, or create and delete catalog permits.

User Action: Ask the owner to perform the requested operation.

--ERROR PF 1385-- {text} is expected to be the name of a subcatalog, but it is a permanent file.

Condition Identifier: PFE\$NTH_NAME_NOT_SUBCATALOG Code: PF 1385

Description: The specified name is the name of a permanent file and not a subcatalog.

User Action: Correct the path.

--ERROR PF 1525-- File {text} resides in a catalog that was restored. Your file attributes, file permits, access log, modification date time, or access date time may be inaccurate. Use change_catalog_entry delete_damage_condition=parent_catalog_restored to clear this error.

Condition Identifier: PFE\$PARENT_CATALOG_RESTORED Code: PF 1525

Description: The device on which the parent catalog of the file resides has been lost and the catalog information was restored from a backup. This catalog information include the file attributes, file permits, access log, modification date time, and access date time. If this information was modified after the backup was made, that modification was lost.

User Action: Use the CHANGE_CATALOG_ENTRY command with DELETE_DAMAGE_CONDITIONS=PARENT_CATALOG_RESTORED to clear this error from the file.

--ERROR PF 1533-- The "{text}" parameter requires at least {text} names; {text} was given.

Condition Identifier: PFE\$PATH_TOO_SHORT Code: PF 1533

Description: The path was too short for the requested operation.

User Action: Correct the path to be of the proper length.

--ERROR PF 1566-- A permanent file system error has occurred. Print the job log and submit it along with a programming system report (PSR). {text} {text} {text} {text} {text} {text}

Condition Identifier: PFE\$PF_SYSTEM_ERROR Code: PF 1566

Description: An internal inconsistency was detected.

User Action: Contact the site analyst. Print the job's dayfile and submit a programming system report (PSR).

--ERROR PF 1750-- File {text} has been restored. Changes made between {text} {text} and a failure have been lost. Use change_catalog_entry delete_damage_condition=respf_modification_mismatch to clear this error.

Condition Identifier: PFE\$RESPF_MODIFICATION_MISMATCH Code: PF 1750

Description: The device on which the cycle resides has been lost, and the cycle data was restored from a backup. The data was modified since the time of the backup, so some work was lost.

User Action: Use CHANGE_CATALOG_ENTRY DELETE_DAMAGE_CONDITION=RESPF_MODIFICATION_MISMATCH to clear this error. The cycle will then be attachable.

--ERROR PF 1850-- To access permanent file {text}, you are required to allow sharing of {text}, not the intended sharing of {text}.

Condition Identifier: PFE\$SHARING_NOT_PERMITTED Code: PF 1850

Description: Access to the specified permanent file for the requested sharing is not permitted.

User Action: Either ask the file owner to grant you greater permission to the specified file, or specify at least the share modes specified in the status message text.

--ERROR PF 1950-- A path may be composed of no more than {text} catalogs. {text} catalogs were specified in path {text}.

Condition Identifier: PFE\$TOO_MANY_CATALOGS_IN_PATH Code: PF 1950

Description: A path containing more than 100 catalog names was specified.

User Action: Specify a path containing no more than 100 catalog names.

--ERROR PF 2011-- No data is defined for cycle {text} of permanent file {text}.

Condition Identifier: PFE\$UNDEFINED_DATA Code: PF 2011

Description: A RESTORE_PERMANENT_FILE was done, which did not restore the cycle data for the specified file.

User Action: Delete the file cycle and restore the existing file.

--ERROR PF 2022-- Cycle number "{text}" of permanent file {text} is unknown.

Condition Identifier: PFE\$UNKNOWN_CYCLE Code: PF 2022

Description: The cycle with the specified cycle number of the specified permanent file is unknown.

User Action: Use DISPLAY_CATALOG_ENTRY to determine the correct cycle number.

--ERROR PF 2033-- Family name {text} is unknown.

Condition Identifier: PFE\$UNKNOWN_FAMILY Code: PF 2033

Description: The specified family name is unknown.

User Action: Specify the correct family name. If the current job's family is desired, OSC\$NULL_NAME can be specified at the program interface level or \$FAMILY at the command level.

--ERROR PF 2044-- Subcatalog name {text} is unknown.

Condition Identifier: PFES\$UNKNOWN_LAST_SUBCATALOG Code: PF 2044

Description: The last subcatalog in the path is unknown.

User Action: Use DISPLAY_CATALOG to determine the correct subcatalog name.

--ERROR PF 2055-- Master catalog name {text} is unknown.

Condition Identifier: PFES\$UNKNOWN_MASTER_CATALOG Code: PF 2055

Description: The specified master catalog name is unknown.

User Action: Specify the correct user name. If the current job's user is desired, OSC\$NULL_NAME may be specified at the program interface level or \$USER at the command level.

--ERROR PF 2066-- Subcatalog name {text} is unknown.

Condition Identifier: PFES\$UNKNOWN_NTH_SUBCATALOG Code: PF 2066

Description: The indicated subcatalog name in the path is unknown.

User Action: Use DISPLAY_CATALOG to determine the correct subcatalog name.

--ERROR PF 2077-- File {file} does not exist or you are not permitted for any access.

Condition Identifier: PFES\$UNKNOWN_PERMANENT_FILE Code: PF 2077

Description: The specified file name is either unknown or you are not permitted access to the file.

User Action: Correct the command to specify the correct path. Determine why the file does not exist, or ask the owner of the file to grant you permission to the file.

--ERROR PF 2088-- You are only permitted access to permanent file {text} for {text} usage, not the requested usage of {text}.

Condition Identifier: PFES\$USAGE_NOT_PERMITTED Code: PF 2088

Description: Access to the specified permanent file for the requested usage is not allowed.

User Action: Either ask the file owner to grant you greater ACCESS_MODE permission to the specified file, or specify only the access permitted as specified in the status message text.

--ERROR PF 2095-- Volume {text} not on-line for file {file}.

Condition Identifier: PFES\$VOLUME_NOT_ONLINE Code: PF 2095

Description: The device on which the file resides has been removed from the active configuration.

User Action: Wait for the device to be added back into the active configuration or for the file to be restored to an active device.

--ERROR PF 2099-- Volume {text} unavailable for file {file}.

Condition Identifier: PFES\$VOLUME_UNAVAILABLE Code: PF 2099

Description: The device on which the file resides is temporarily unavailable.

User Action: Wait for the device to become available or for the file to be restored to an available device.

--ERROR PF 5325-- Data is already defined for cycle {text} of permanent file {text}.

Condition Identifier: PFES\$DATA_ALREADY_DEFINED Code: PF 5325

Description: An attempt was made to restore a cycle for which data already exists on the active disks.

User Action: Either ignore the error because the cycle already exists, or if the data is desired from the backup file rather than the active disks, either delete the cycle and repeat the restore, or restore the data to a new file.

--WARNING PF 6733-- {text} {text} occurred during permanent file recovery. {text} {text}

Condition Identifier: PFE\$RECOVERY_SUMMARY Code: PF 6733

Description: This message is informative only, unless the number of errors displayed is non-zero, indicating an internal error.

User Action: Contact your site analyst if the number of errors is non-zero. The system and job logs will also contain error messages.

--ERROR PF 6782-- File: {text} was not selected for restore by RESTORE_EXCLUDED_FILE_CYCLES.

Condition Identifier: PFE\$RESTORE_NOT_SELECTED Code: PF 6782

Description: The file was not restored because the RESTORE_OPTIONS specified did not include NO_DATA_DEFINED.

User Action: Specify RESTORE_OPTIONS=NO_DATA_DEFINED on the call to RESTORE_EXCLUDED_FILE_CYCLES.

--WARNING PM 0-- String {text} is not a valid {text} file name.

Condition Identifier: PME\$INVALID_FILE_NAME Code: PM 0

Description: The program description contains an invalid file name.

User Action: Correct the invalid file name.

--ERROR PM 1-- Specified value for TERMINATION_ERROR_LEVEL is unknown.

Condition Identifier: PME\$INVALID_TERM_ERROR_LEVEL Code: PM 1

Description: The valid termination error levels are warning, error, and fatal.

User Action: Specify a valid termination error level.

--ERROR PM 2-- Specified value for PRESET is unknown.

Condition Identifier: PME\$INVALID_PRESET_OPTION Code: PM 2

Description: The valid preset options are preset to zero, to all ones, to the indefinite value, or to the infinity value.

User Action: Specify a valid preset option.

--ERROR PM 3-- Specified MAXIMUM_STACK_SIZE exceeds maximum segment length.

Condition Identifier: PME\$INVALID_STACK_SIZE_OPTION Code: PM 3

Description: The specified maximum stack size must be within the valid range for the length of a segment.

User Action: Correct the maximum stack size value.

--ERROR PM 4-- Specified LOAD_MAP_OPTIONS contains NONE and another option.

Condition Identifier: PME\$MAP_OPTION_CONFLICT Code: PM 4

Description: The keyword value NONE requests that no load map be written. Specification of any other option requests a load map.

User Action: Either specify NONE alone or specify one or more of the other load map options.

--ERROR PM 5-- Specified WAIT parameter is not OSC\$WAIT or OSC\$NO_WAIT.

Condition Identifier: PME\$INVALID_WAIT_PARAMETER Code: PM 5

Description: If you specify OSC\$WAIT, the task waits for completion; if you specify OSC\$NOWAIT, the task does not wait.

User Action: Specify a valid wait option identifier.

--ERROR PM 10-- Library list change kind unknown.

Condition Identifier: PME\$INVALID_LIBRARY_CHANGE_KIND Code: PM 10

Description: The library list change kind is unknown.

User Action: Specify a valid library list change.

--ERROR PM 11-- No libraries specified in ADD or DELETE request.

Condition Identifier: PME\$NO_LIBRARIES_IN_CHANGE Code: PM 11

Description: No libraries are specified in the ADD or DELETE request.

User Action: Specify a valid library identifier.

--ERROR PM 12-- Library list currently empty.

Condition Identifier: PME\$EMPTY_LIBRARY_LIST Code: PM 12

Description: The library list is currently empty.

User Action: None.

--WARNING PM 13-- Library {file} not present in library list.

Condition Identifier: PME\$UNKNOWN_DELETE_LIBRARY Code: PM 13

Description: The specified library file is not present in the library list.

User Action: Add the specified library file to the library list.

--WARNING PM 14-- Library {file} already present in library list.

Condition Identifier: PME\$DUPLICATE_ADD_LIBRARY Code: PM 14

Description: The library file is already on the library list.

User Action: Do not attempt to re-enter this file on the library list.

--ERROR PM 15-- Output list has different dimension than current library list.

Condition Identifier: PME\$LIBRARY_COUNT_MISMATCH Code: PM 15

Description: The output list is not the same length as the current library list.

User Action: Change the output list length to match the library list length.

--WARNING PM 16-- {file} is not an existing library.

Condition Identifier: PME\$FILE_NOT_EXISTING_LIBRARY Code: PM 16

Description: The specified file does not contain an object library.

User Action: Check that the correct file was specified.

--WARNING PM 17-- {file} cannot be an existing file.

Condition Identifier: PME\$RESERVED_LIBRARY_NAME Code: PM 17

Description: The specified file cannot be an existing file.

User Action: The specified file must be deleted.

--ERROR PM 18-- Some libraries were not {text}.

Condition Identifier: PME\$SOME_LIBRARIES_NOT Code: PM 18

Description: Some libraries were already added to or deleted from the library list.

User Action: The specified libraries do not have to be respecified.

--ERROR PM 20-- Program description is too small to contain {text}.

Condition Identifier: PME\$PROG_DESCRIPTION_TOO_SMALL Code: PM 20

Description: The specified program description variable is too small to contain the program description.

User Action: Increase the size of the program description variable allocated.

--ERROR PM 21-- Specified number of {text} is 0 or exceeds {text}.

Condition Identifier: PMS\$INVALID_LIST_LENGTH Code: PM 21

Description: A list length within the program description is not within the valid range of values.

User Action: Correct the invalid list length.

--ERROR PM 22-- Initial program transfer may not be to ring 1 or 2.

Condition Identifier: PMS\$TRANSFER_ADDRESS_RING_ERROR Code: PM 22

Description: The initial program transfer may not be to ring 1 or 2.

User Action: Transfer the initial program to another ring.

--ERROR PM 23-- Task id not within valid range.

Condition Identifier: PMS\$INVALID_TASK_ID Code: PM 23

Description: The task identifier specified must be the task identifier returned by the PMP\$EXECUTE call that started the task.

User Action: Specify the returned task identifier.

--ERROR PM 24-- Specified task is not a child of executing task.

Condition Identifier: PMS\$TASK_NOT_CURRENT_CHILD Code: PM 24

Description: The task was not started by the requesting task.

User Action: Issue the call to terminate the started task from the starting task.

--ERROR PM 25-- Task termination requested by parent task.

Condition Identifier: PMS\$TERMINATED_BY_PARENT Code: PM 25

Description: Task termination has been requested by the parent task.

User Action: Determine why the parent task requested termination.

--ERROR PM 26-- Size of parameter not equal to size of current program description.

Condition Identifier: PMS\$PROG_DESCRIPTION_SIZE_ERROR Code: PM 26

Description: The program description variable allocated should be the length returned by a PMP\$GET_PROGRAM_SIZE call.

User Action: Call PMP\$GET_PROGRAM_SIZE to determine the length of the program description of the calling task and allocation a program description variable of that length.

--ERROR PM 40-- Stack destroyed - some block exit handlers may not have been activated.

Condition Identifier: PMS\$STACK_FRAME_POPPER_ABORTED Code: PM 40

Description: The stack has been destroyed. Some block exit handlers may not have been activated.

User Action: Activate needed block exit handlers.

--ERROR PM 50-- Program termination is beyond the point of revocation.

Condition Identifier: PMS\$TERMINATION_NOT_REVOCABLE Code: PM 50

Description: It is too late to revoke program termination.

User Action: None.

--ERROR PM 51-- Maximum program termination revocations exceeded.

Condition Identifier: PMS\$MAXIMUM_TERM_REVOCATIONS Code: PM 51

Description: The last attempt to revoke program termination exceeded the maximum limit.

User Action: No more program termination revocations may be done.

--ERROR PM 60-- Common block {text} not unallocated common.

Condition Identifier: PME\$COMMON_NOT_UNALLOCATED Code: PM 60

Description: The common block named is not an unallocated common block.

User Action: Specify an unallocated common block for association with the file.

--ERROR PM 61-- File to be opened for common block {text} is already open.

Condition Identifier: PME\$COMMON_FILE_OPEN Code: PM 61

Description: The file associated with the named unallocated common block is already open.

User Action: Close the file before attempting to open it again.

--ERROR PM 62-- The file for common block {text} is not open.

Condition Identifier: PME\$COMMON_FILE_NOT_OPEN Code: PM 62

Description: The file associated with the named unallocated common block is not open.

User Action: Do not attempt to close a file that is not open.

--ERROR PM 4900-- Task_id not currently assigned to any active task.

Condition Identifier: PME\$UNKNOWN_TASK_ID Code: PM 4900

Description: The task identifier is not currently assigned to any active task.

User Action: Correct the task identifier.

--ERROR PM 5002-- Condition name, {text}, is improper.

Condition Identifier: PME\$INCORRECT_CONDITION_NAME Code: PM 5002

Description: Each condition name specified must be valid for one of the condition categories selected by the condition_selector field.

User Action: Correct the invalid condition name.

--ERROR PM 5003-- Establish descriptor is not an automatic variable of the requestor.

Condition Identifier: PME\$DESCRIPTOR_ADDRESS_ERROR Code: PM 5003

Description: The establish_descriptor parameter variable cannot be declared with an attribute.

User Action: Remove the attribute specification from the descriptor variable declaration.

--ERROR PM 5004-- No handler is in effect within the scope of the condition.

Condition Identifier: PME\$NO_ESTABLISHED_HANDLER Code: PM 5004

Description: The requested operation cannot be performed because no condition handler is established for the specified condition set.

User Action: Establish a condition handler for the specified condition set.

--ERROR PM 5005-- A stack segment contains invalid frames.

Condition Identifier: PME\$INCONSISTENT_STACK Code: PM 5005

Description: Probable internal system error.

User Action: Notify a site analyst of the error, providing appropriate documentation.

--ERROR PM 5006-- A stack of established condition handlers is erroneous.

Condition Identifier: PME\$HANDLER_STACK_ERROR Code: PM 5006

Description: Probable internal system error.

User Action: Notify a site analyst of the error, providing appropriate documentation.

--FATAL PM 5007-- The condition handler in effect for {text} is not callable.

Condition Identifier: PME\$INVALID_CONDITION_HANDLER Code: PM 5007

Description: The system was unable to call the condition handler for the task.

User Action: Ensure that the condition handler is callable.

--FATAL PM 5008-- A condition handler caused the same condition the handler was processing.

Condition Identifier: PME\$CONDITION_IN_HANDLER Code: PM 5008

Description: The call is valid only within a condition handler established for the condition; the condition handler must have been called due to an occurrence of the condition.

User Action: Establish the procedure as a condition handler. To test the condition handler, call PMP\$TEST_CONDITION_HANDLER.

--ERROR PM 5009-- PMP\$CONTINUE_TO_CAUSE requested from outside the realm of a condition handler.

Condition Identifier: PME\$NO_CONDITION_TO_CONTINUE Code: PM 5009

Description: PMP\$CONTINUE_TO_CAUSE can only be called from within a condition handler.

User Action: If possible, find another method to cause the condition.

--ERROR PM 5010-- Invalid condition selector - establish or disestablish.

Condition Identifier: PME\$INVALID_CONDITION_SELECTOR Code: PM 5010

Description: The condition_selector field specifies all condition categories, a combination of condition categories, or a single condition category.

User Action: Correct the condition_selector identifier.

--ERROR PM 5011-- A specified system condition is improper for the request.

Condition Identifier: PME\$UNSELECTABLE_CONDITION Code: PM 5011

Description: Although the condition identifier is valid, the specified condition cannot be selected for condition processing or its detection enabled or inhibited.

User Action: Remove the condition identifier from the call.

--ERROR PM 5012-- Attempted to continue a condition to a handler active processing that condition.

Condition Identifier: PME\$RECURSIVE_CONTINUE Code: PM 5012

Description: The next condition handler has already been called to process the condition. The condition handler was established for more than one condition set that included the condition.

User Action: Ensure that each condition handler is established only once for a condition type.

--ERROR PM 5013-- Standard procedure selection is improper.

Condition Identifier: PME\$INVALID_STANDARD_SELECTION Code: PM 5013

Description: The valid selection identifiers are PMC\$EXECUTE_STANDARD_PROCEDURE and PMC\$INHIBIT_STANDARD_PROCEDURE.

User Action: Correct the selection identifier.

--ERROR PM 5014-- Ring={text} stack segment overwritten.

Condition Identifier: PME\$STACK_OVERWRITTEN Code: PM 5014

Description: Probable internal system error.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR PM 5015-- PMP\$TEST_CONDITION_HANDLER does not support the specified condition.

Condition Identifier: PME\$UNSUPPORTED_BY_TEST_COND Code: PM 5015

Description: The condition set included a condition that cannot be forced by a PMP\$TEST_CONDITION_HANDLER call.

User Action: Remove the unsupported condition.

--ERROR PM 5020-- PMP\$SET_PROCESS_INTERVAL_TIMER microsecond parameter is out of range.

Condition Identifier: PME\$PIT_VALUE_OUT_OF_RANGE Code: PM 5020

Description: The valid interval range is 1000 to 07ffffff(16).

User Action: Correct the specified interval value.

--ERROR PM 5050-- Maximum number of queues are already defined in the job.

Condition Identifier: PME\$MAXIMUM_QUEUES_DEFINED Code: PM 5050

Description: The maximum defined queues allowed is determined by the value of the constant PMC\$MAX_QUEUES_PER_JOB.

User Action: Define a queue only when the number of queues defined is less than the maximum. To determine the number of queues currently defined, call PMP\$STATUS_QUEUES_DEFINED.

--ERROR PM 5051-- The queue, {text}, is already defined.

Condition Identifier: PME\$QUEUE_ALREADY_DEFINED Code: PM 5051

Description: The queue cannot be defined because a queue by that name is currently defined.

User Action: Either use the current queue definition or remove the queue definition and define a new queue using that name.

--ERROR PM 5052-- Attempted to remove {text} from a ring greater than the removal bracket.

Condition Identifier: PME\$REMOVAL_BRACKET_ERROR Code: PM 5052

Description: The task attempting to remove the queue must execute within a ring less than or equal to the removal bracket ring specified when the queue was defined.

User Action: Either change the removal bracket ring in the queue definition or change the ring in which the task executes.

--ERROR PM 5053-- Usage bracket more privileged than removal bracket for {text}.

Condition Identifier: PME\$USAGE_LT_REMOVAL_BRACKET Code: PM 5053

Description: The specified usage bracket ring must be greater than or equal to the specified removal bracket ring.

User Action: Either increase the specified usage bracket or decrease the specified removal bracket.

--ERROR PM 5054-- Queue, {text}, is undefined.

Condition Identifier: PME\$UNKNOWN_QUEUE_NAME Code: PM 5054

Description: The specified queue name does not identify a queue currently defined within the job.

User Action: Ensure that the specified name is correct and that the queue has been defined.

--ERROR PM 5055-- The queue, {text}, has tasks connected.

Condition Identifier: PME\$TASKS_CONNECTED_TO_QUEUE Code: PM 5055

Description: The queue definition cannot be removed while tasks are connected to the queue.

User Action: Wait until the tasks have disconnected from the queue. To determine the number of connected tasks, call PMP\$STATUS_QUEUE.

--ERROR PM 5056-- The queue, {text}, contains messages.

Condition Identifier: PME\$NONEMPTY_QUEUE Code: PM 5056

Description: The queue definition cannot be removed until all messages in the queue have been received.

User Action: Wait until all messages are received or receive the messages with PMP\$RECEIVE_FROM_QUEUE calls. To determine the number of queued messages, call PMP\$STATUS_QUEUE.

--ERROR PM 5057-- Maximum number of tasks are already connected to {text}.

Condition Identifier: PME\$MAXIMUM_TASKS_CONNECTED Code: PM 5057

Description: The maximum connected tasks allowed is determined by the value of the constant PMC\$MAX_CONNECTED_PER_QUEUE.

User Action: Connect a task to a queue only when the number of connected tasks is less than the maximum. To determine the number of tasks currently connected, call PMP\$STATUS_QUEUE.

--ERROR PM 5058-- Task is already connected to {text}.

Condition Identifier: PME\$TASK_ALREADY_CONNECTED Code: PM 5058

Description: The task cannot connect to the queue because it is already connected to the queue.

User Action: Either use the current connection or disconnect the task from the queue and request a new connection.

--ERROR PM 5059-- Requestor has insufficient privilege for {text}.

Condition Identifier: PME\$USAGE_BRACKET_ERROR Code: PM 5059

Description: The ring within which the task is executing is greater than the usage bracket ring specified when the queue was defined.

User Action: Either increase the usage bracket ring specified on the queue definition or decrease the ring in which the task is executing.

--ERROR PM 5060-- Task is not connected to {text}.

Condition Identifier: PME\$UNKNOWN_QUEUE_IDENTIFIER Code: PM 5060

Description: The specified queue identifier does not identify a current queue connection.

User Action: Specify the queue identifier returned by the PMP\$CONNECT_QUEUE call.

--ERROR PM 5061-- Maximum number of messages are already on {text}.

Condition Identifier: PME\$MAXIMUM_QUEUED_MESSAGES Code: PM 5061

Description: The maximum queued messages allowed is determined by the value of the constant PMC\$MAX_MESSAGES_PER_QUEUE.

User Action: Send a message only when the number of queued messages is less than the maximum. To determine the number of messages currently queued, call PMP\$STATUS_QUEUE.

--ERROR PM 5062-- Passing or sharing executable, binding or stack segment is not allowed.

Condition Identifier: PME\$INCORRECT_SEGMENT_MESSAGE Code: PM 5062

Description: Currently, a queue cannot pass a segment pointer or share segments.

User Action: None.

--ERROR PM 5063-- Incorrect queue message type.

Condition Identifier: PME\$INCORRECT_MESSAGE_TYPE Code: PM 5063

Description: Currently, the only valid message kind is PMC\$MESSAGE_VALUE.

User Action: Correct the message kind specification.

--ERROR PM 5064-- Incorrect queued segment type.

Condition Identifier: PME\$INCORRECT_QUEUED_SEG_TYPE Code: PM 5064

Description: Currently, a queue cannot pass a segment pointer or share segments.

User Action: None.

--ERROR PM 5065-- Requestor has insufficient privilege to access the pointer object.

Condition Identifier: PME\$ERROR_POINTER_PRIVILEGE Code: PM 5065

Description: Currently, a queue cannot pass a segment pointer or share segments.

User Action: None.

--ERROR PM 5066-- Requestor has insufficient privilege to access a passed/shared segment.

Condition Identifier: PME\$ERROR_SEGMENT_PRIVILEGE Code: PM 5066

Description: Currently, a queue cannot pass a segment pointer or share segments.

User Action: None.

--ERROR PM 5067-- Number_of_segments to be passed/shared is not in the valid range.

Condition Identifier: PME\$ERROR_NUMBER_OF_SEGMENTS Code: PM 5067

Description: Currently, a queue cannot pass a segment pointer or share segments.

User Action: None.

--ERROR PM 5068-- The maximum number of tasks are already waiting for a message on {text}.

Condition Identifier: PME\$MAXIMUM_WAITING_TASKS Code: PM 5068

Description: The maximum number of tasks is already waiting for a message.

User Action: The task must wait until one of the waiting tasks receives a message.

--ERROR PM 5069-- Queue name, {text}, is improper.

Condition Identifier: PME\$INCORRECT_QUEUE_NAME Code: PM 5069

Description: The specified queue name is not a valid name.

User Action: Specify a valid name as the queue name.

--ERROR PM 5070-- Requestor is less privileged than the removal bracket for {text}.

Condition Identifier: PME\$REQUEST_GT_REMOVAL_BRACKET Code: PM 5070

Description: The specified removal bracket ring cannot be more privileged than the ring in which the queue is defined.

User Action: Increase the specified removal bracket ring to at least the ring in which the defining task executes.

--ERROR PM 5071-- Segment passing/sharing is not currently supported.

Condition Identifier: PME\$PASS_SHARE_PROHIBITED Code: PM 5071

Description: Currently, a queue cannot pass a segment pointer or share a segment.

User Action: None.

--ERROR PM 5072-- Caller is outside the removal bracket for {text}.

Condition Identifier: PME\$CALLER_GT_REMOVAL_BRACKET Code: PM 5072

Description: The task which defines the queue is executing in a ring greater than the specified removal ring.

User Action: Change the removal ring.

--FATAL PM 5100-- {text} at P={text}.

Condition Identifier: PME\$SYSTEM_CONDITION Code: PM 5100

Description: The specified condition occurred at the specified program virtual address.

User Action: If possible, determine if the contents of the specified address are in error.

--FATAL PM 5101-- Unknown system request at P={text}.

Condition Identifier: PME\$UNKNOWN_SYSTEM_REQUEST Code: PM 5101

Description: An invalid system request was made from the specified address.

User Action: Investigate the specified address.

--INFORMATIVE PM 5102-- Task Name: {text}. Global Task ID: {text}.

Condition Identifier: PME\$TASK_BEGIN_INFORMATION Code: PM 5102

Description: This message contains the task name of a task at task begin.

User Action: None.

--INFORMATIVE PM 5103-- Task Name: {text}. Global Task ID: {text}. Completion Status: {text}.

Condition Identifier: PME\$TASK_END_INFORMATION Code: PM 5103

Description: This message contains the task name of a task and its termination status.

User Action: None.

--ERROR PM 5150-- incorrect date format

Condition Identifier: PME\$INVALID_DATE_FORMAT Code: PM 5150

Description: The valid date format identifiers are OSC\$MONTH_DATE, OSC\$MDY_DATE, OSC\$ISO_DATE, OSC\$ORDINAL_DATE, OSC\$DMY_DATE, and OSC\$DEFAULT_DATE.

User Action: Correct the date format identifier.

--ERROR PM 5151-- incorrect time format

Condition Identifier: PME\$INVALID_TIME_FORMAT Code: PM 5151

Description: The valid time format identifiers are OSC\$AMPM_TIME, OSC\$HMS_TIME, OSC\$MILLISECOND_TIME, and OSC\$DEFAULT_TIME.

User Action: Correct the date format identifier.

--ERROR PM 5152-- invalid spy identifier

Condition Identifier: PME\$INVALID_SPY_IDENTIFIER Code: PM 5152

Description: The spy identifier is invalid.

User Action: Change the spy identifier to a valid value.

--ERROR PM 5153-- common block {text} not defined

Condition Identifier: PME\$COMMON_BLOCK_NOT_DEFINED Code: PM 5153

Description: The specified common block is not defined.

User Action: Check the load map or source code to see what common blocks exist.

--ERROR PM 5154-- library header missing

Condition Identifier: PME\$LIBRARY_HEADER_MISSING Code: PM 5154

Description: The library header is missing from the object library.

User Action: Regenerate the object library using the original modules.

--ERROR PM 5155-- invalid library version {text}

Condition Identifier: PME\$WRONG_LIBRARY_VERSION Code: PM 5155

Description: The library version is invalid.

User Action: Regenerate the library using the original modules.

--ERROR PM 5156-- bad entry point dictionary pointer encountered

Condition Identifier: PME\$BAD_ENTRY_DICTIONARY_PTR Code: PM 5156

Description: A bad entry point dictionary pointer has been encountered on the object library.

User Action: Regenerate the object library using the original modules.

--ERROR PM 5157-- The result array {text} is not large enough to contain the requested information.

Condition Identifier: PME\$RESULT_ARRAY_TOO_SMALL Code: PM 5157

Description: The array specified on the request is not large enough to contain the data requested.

User Action: Retry the request with a larger array. Most interfaces will return an indication of how large the array should be.

--ERROR PM 5158-- unknown entry point {text}

Condition Identifier: PME\$UNKNOWN_ENTRY_POINT Code: PM 5158

Description: The named entry point is not defined in the task.

User Action: Load the module containing the entry point.

--ERROR PM 5200-- Missing module definition.

Condition Identifier: PME\$MISSING_MODULE_DEFINITION Code: PM 5200

Description: A module definition is missing from the object module.

User Action: Regenerate the object module using the original source.

--ERROR PM 5201-- Module not terminated: {text}.

Condition Identifier: PME\$MISSING_MODULE_TERMINATION Code: PM 5201

Description: The specified object module does not have a transfer symbol record.

User Action: Regenerate the object module from the original source.

--ERROR PM 5202-- Internal module segment overflow: {text}.

Condition Identifier: PME\$MODULE_SEGMENT_OVERFLOW Code: PM 5202

Description: One of the Debug Table builder's scratch files has overflowed.

User Action: Inform the site analyst providing appropriate documentation of the error.

--ERROR PM 5203-- Internal entry point segment overflow: {text}.

Condition Identifier: PME\$ENTRY_PT_SEGMENT_OVERFLOW Code: PM 5203

Description: One of the debug table builder's scratch files has overflowed.

User Action: Inform a site analyst providing appropriate documentation of the error.

--ERROR PM 5204-- Bad symbol table fragment encountered on module: {text}.

Condition Identifier: PME\$BAD_SYMBOL_TABLE_FRAGMENT Code: PM 5204

Description: The specified module contains an invalid symbol table.

User Action: Regenerate the module from the original source.

--ERROR PM 5205-- Undefined debug id.

Condition Identifier: PME\$UNDEFINED_DEBUG_ID Code: PM 5205

Description: The Debug identifier is undefined.

User Action: Redefine the Debug identifier.

--ERROR PM 5206-- Debug entry list overflow - entry ignored.

Condition Identifier: PME\$TOO_MANY_DEBUG_LIST_ENTRIES Code: PM 5206

Description: The entry is ignored because the debug entry list has overflowed.

User Action: Delete a breakpoint.

--ERROR PM 5207-- No debug codes specified.

Condition Identifier: PME\$EMPTY_DEBUG_CODE Code: PM 5207

Description: No debugging codes are specified.

User Action: Specify a debugging code for the break point.

--ERROR PM 5208-- Segments unequal in debug addresses.

Condition Identifier: PME\$ADDRESS_SEGMENTS_NOT_EQUAL Code: PM 5208

Description: The range specified for the breakpoint contains unequal segment numbers.

User Action: Equalize the segments.

--ERROR PM 5209-- Low debug address is greater than high debug address.

Condition Identifier: PME\$LOW_ADDR_GREATER_THAN_HIGH Code: PM 5209

Description: The lower offset specified for the breakpoint is greater than the upper offset.

User Action: Respecify the range in the correct order.

--ERROR PM 5210-- Address specified can not be accessed.

Condition Identifier: PME\$INVALID_ACCESS Code: PM 5210

Description: The specified address cannot be accessed for a breakpoint.

User Action: Respecify the address.

--ERROR PM 5211-- Undefined debug index.

Condition Identifier: PME\$UNDEFINED_DEBUG_INDEX Code: PM 5211

Description: The Debug index is undefined.

User Action: Redefine the Debug index.

--ERROR PM 5212-- Stack frame not found.

Condition Identifier: PME\$STACK_FRAME_NOT_FOUND Code: PM 5212

Description: The specified stack frame could not be found.

User Action: Correct the stack frame address.

--ERROR PM 5213-- Too many entry points.

Condition Identifier: PME\$TOO_MANY_ENTRY_POINTS Code: PM 5213

Description: Internal system error. Too many entry points are defined.

User Action: Inform a site analyst, providing appropriate documentation of the error.

--ERROR PM 5214-- Invalid section ordinal: {text}.

Condition Identifier: PME\$INVALID_SECTION_ORDINAL Code: PM 5214

Description: A module format error has occurred.

User Action: Regenerate the module by recompiling its source code. If the problem recurs, inform a site analyst.

--ERROR PM 5215-- Invalid ring number: {text}.

Condition Identifier: PME\$INVALID_RING_NUMBER Code: PM 5215

Description: The specified ring number is invalid.

User Action: Check the load map for diagnostics.

--ERROR PM 5217-- Invalid line address table encountered.

Condition Identifier: PME\$INVALID_LINE_ADDRESS_TABLE Code: PM 5217

Description: A module format error has occurred.

User Action: Regenerate the module by recompiling its source code. If the problem recurs, inform a site analyst.

--WARNING PM 5218-- Unable to load interactive debugger entry point {text}.

Condition Identifier: PME\$UNABLE_TO_LOAD_DEBUG Code: PM 5218

Description: Due to load errors, the specified Debug entry point could not be loaded.

User Action: Check the load map for diagnostics and correct.

--ERROR PM 5219-- Bad debug symbol table encountered on module: {text}.

Condition Identifier: PME\$BAD_DEBUG_SYMBOL_TABLE Code: PM 5219

Description: A module format error has occurred.

User Action: Regenerate the module by recompiling its source code. If the problem recurs, inform a site analyst.

--ERROR PM 5250-- Logging has not been initialized.

Condition Identifier: PME\$LOGGING_NOT_YET_ACTIVE Code: PM 5250

Description: A log message has been issued before the log has been initialized.

User Action: Wait for the log to be initialized before issuing log messages.

--ERROR PM 5251-- Caller attempted to add to the JOB_LOG after the JOB_LOG had appended to the output file.

Condition Identifier: PME\$JOB_LOG_NO_LONGER_ACTIVE Code: PM 5251

Description: A log message was issued when the JOB LOG was no longer active.

User Action: Issue a message to the log before JOB_LOG is appended to the output file.

--INFORMATIVE PM 5252-- Caller does not have sufficient privilege to write in the SYSTEM_LOG.

Condition Identifier: PME\$CANNOT_WRITE_IN_SYSTEM_LOG Code: PM 5252

Description: The caller has insufficient ring privileges to write to the SYSTEM_LOG.

User Action: Do not attempt to write to the system log.

--ERROR PM 5253-- The log cycle provided by the caller does not match the current cycle for log {text}.

Condition Identifier: PME\$LOG_CYCLES_DO_NOT_MATCH Code: PM 5253

Description: The log cycle provided by the caller does not match the current cycle for the specified log.

User Action: Provide the correct log cycle.

--ERROR PM 5254-- The file identifier passed does not represent a log file.

Condition Identifier: PME\$FILE_ID_IS_NOT_LOG_FILE Code: PM 5254

Description: The file identifier passed does not reference a log file.

User Action: Supply a valid file identifier.

--ERROR PM 5255-- Caller does not have the necessary privilege on log {text} to {text}.

Condition Identifier: PME\$REQUESTED_OP_NOT_ALLOWED Code: PM 5255

Description: The caller lacks the necessary privileges to perform the specified operation on the log.

User Action: None.

--ERROR PM 5256-- Not enough space in wsa ({text}) for log entry of length ({text}).

Condition Identifier: PME\$NOT_ENOUGH_SPACE_IN_WSA Code: PM 5256

Description: The issued message is too long for a log entry.

User Action: Reissue a shorter message.

--ERROR PM 5257-- The local file name ({text}) passed in the OPEN request does not match a log name.

Condition Identifier: PME\$NAME_IS_NOT_NAME_OF_LOG Code: PM 5257

Description: The specified local file name passed in the OPEN request does not match a log name.

User Action: Reissue the request with a valid log name.

--ERROR PM 5258-- The requested operation is not defined for an active log.

Condition Identifier: PME\$REQUESTED_OP_NOT_DEFINED Code: PM 5258

Description: The requested operation is not defined for an active log.

User Action: Do not issue operations on an active file.

--ERROR PM 5259-- Specified new boi address is beyond the current EOI.

Condition Identifier: PME\$NEW_BOI_ADDR_OUT_OF_RANGE Code: PM 5259

Description: Specified beginning of information address cannot be beyond the end of the information address of the log file.

User Action: Reissue request with a valid beginning of information address.

--ERROR PM 5261-- Specified backup count was {text} (limit is {text}).

Condition Identifier: PME\$BACKUP_LIMIT_EXCEEDED Code: PM 5261

Description: The specified backup count exceeds the maximum limit.

User Action: Issue a request with a valid backup count.

--ERROR PM 5262-- If the FROM parameter is LAST, then the CONTENT parameter must be an integer.

Condition Identifier: PME\$CONTENT_PARAM_NOT_INTEGER Code: PM 5262

Description: The CONTENT parameter must be an integer when the FROM parameter is LAST.

User Action: Reissue the command.

--ERROR PM 5263-- If the FROM parameter is TIME or MESSAGE, then the CONTENT parameter must be a string.

Condition Identifier: PME\$CONTENT_PARAM_NOT_STRING Code: PM 5263

Description: The CONTENT parameter must be a string when the FROM parameter is TIME..MESSAGE.

User Action: Reissue the command.

--ERROR PM 5264-- The string ({text}) could not be found.

Condition Identifier: PME\$SEARCH_DID_NOT_FIND_MATCH Code: PM 5264

Description: The specified string could not be found.

User Action: Reissue the command.

--ERROR PM 5265-- The record just read in is not long enough to be an ascii log entry. It consists of ... {text}.

Condition Identifier: PME\$RECORD_TOO_SHORT Code: PM 5265

Description: The specified record is too short to be an ASCII log entry.

User Action: Regenerate the log.

--ERROR PM 5266-- The variable supplied by the user in the PFN parameter of the TERMINATE_LOG command was not STRING KIND.

Condition Identifier: PME\$VARIABLE_REF_MUST_BE_STRING Code: PM 5266

Description: The permanent file name parameter variable must be of kind STRING.

User Action: Specify a string variable on the command.

--ERROR PM 5267-- Not enough local space to log entry.

Condition Identifier: PME\$PUSH_FAIL Code: PM 5267

Description: Internal system error. Not enough local space exists to log the entry.

User Action: Inform a site analyst, providing appropriate documentation of the error.

--ERROR PM 5268-- Log is full. See site analyst.

Condition Identifier: PME\$LOG_IS_FULL Code: PM 5268

Description: The log is full.

User Action: See the site analyst.

--ERROR PM 5269-- EOF reached while reading log {text}.

Condition Identifier: PME\$END_OF_LOG Code: PM 5269

Description: A premature end of file was encountered on the specified log.

User Action: Regenerate the log.

--ERROR PM 5270-- Local log parameter specified is not a local log.

Condition Identifier: PME\$NOT_LOCAL_LOG Code: PM 5270

Description: The specified local log parameter is not a local log.

User Action: Regenerate the log.

--ERROR PM 5271-- User must be system administrator to terminate log.

Condition Identifier: PME\$NOT_SYSTEM_ADMINISTRATOR Code: PM 5271

Description: The user must be the system administrator to terminate the log.

User Action: Contact the system administrator.

--INFORMATIVE PM 5272-- Caller does not have sufficient privilege to write in the {text}.

Condition Identifier: PME\$CANNOT_WRITE_TO_LOG Code: PM 5272

Description: The user does not have sufficient privilege to write to the LOG.

User Action: See the family administrator about additional privilege.

--ERROR PM 5273-- Metric {text} is specified for display {text}, but is not defined.

Condition Identifier: PME\$UNDEFINED_METRIC Code: PM 5273

Description: The metric name specified on the display subcommand has not been defined.

User Action: Use the DEFINE_METRIC subcommand.

--ERROR PM 5274-- Group {text} is specified for metric {text}, but is not defined.

Condition Identifier: PME\$UNDEFINED_GROUP Code: PM 5274

Description: The group name specified for the metric has not been defined.

User Action: Use the DEFINE_GROUP subcommand.

--ERROR PM 5275-- Group {text} is specified for a group dump, but is not defined.

Condition Identifier: PME\$UNDEFINED_GROUP_FOR_DUMP Code: PM 5275

Description: The group name specified on the group dump subcommand has not been defined.

User Action: Use the DEFINE_GROUP subcommand.

--WARNING PM 5276-- Group {text} was already defined. The new definition is ignored.

Condition Identifier: PME\$REDEFINED_GROUP Code: PM 5276

Description: The specified group name has already been defined.

User Action: None.

--WARNING PM 5277-- Metric {text} was already defined. The new definition is ignored.

Condition Identifier: PME\$REDEFINED_METRIC Code: PM 5277

Description: The specified metric name has already been defined.

User Action: None.

--ERROR PM 5278-- {text} is not a valid statistic name.

Condition Identifier: PME\$BAD_STATISTIC Code: PM 5278

Description: The specified statistic name is not valid.

User Action: See the Statistics Facility Chapter in the NOS/VE System Performance and Maintenance Manual for a list of valid system statistic names.

--ERROR PM 5279-- Some metric type must be specified.

Condition Identifier: PME\$NO_METRIC_TYPE Code: PM 5279

Description: The counter or expression parameter must be used when defining a metric.

User Action: Use either the counter or expression parameter on the DEFINE_METRIC subcommand.

--ERROR PM 5280-- Conflicting metric types specified for metric {text}.

Condition Identifier: PME\$TOO_MANY_METRIC_TYPES Code: PM 5280

Description: Only one metric type, counter or expression, is valid for the definition of a metric.

User Action: Use either the counter or expression parameter on the DEFINE_METRIC subcommand.

--ERROR PM 5281-- A null string may not be specified for descriptive dat3233,

Condition Identifier: PME\$NULL_DESC_DATA Code: PM 5281

Description: A null string may not be specified as a value for the descriptive_data parameter of DEFINE_GROUP.

User Action: Specify a valid string as a value for the descriptive_data parameter.

--ERROR PM 5282-- {text} is not a valid expression.

Condition Identifier: PME\$BAD_EXPRESSION Code: PM 5282

Description: Only the string value of '1' is valid for the expression parameter at this time.

User Action: Specify a string value of 1 for the expression parameter.

--ERROR PM 5283-- {text} is not a legal time

Condition Identifier: PME\$BAD_TIME Code: PM 5283

Description: Time value is not a legal time.

User Action: See the Statistics Facility Chapter in the NOS/VE System Performance and Maintenance Manual for a description of the time parameter on the DEFINE_GROUP subcommand.

--ERROR PM 5284-- {text} low_limit > high_limit in disd subcommand

Condition Identifier: PME\$BAD_HIGH_LOW_LIMIT Code: PM 5284

Description: The low limit is greater than the high limit for the limits parameter on the DISPLAY_DISTRIBUTION subcommand.

User Action: Specify a high limit for the limits parameter that is greater than the low limit.

--ERROR PM 5285-- {text} \$OUTPUT file should not be permanent in gengf subcommand

Condition Identifier: PME\$OUTPUT_PERMANENT Code: PM 5285

Description: The \$OUTPUT file, which is the default value for the OUTPUT parameter, can not be made permanent in the GENGF subcommand.

User Action: Do not specify a value of TRUE for the PERMANENT parameter on the GENERATE_GROUP_FILE when using the default value for the OUTPUT parameter.

--ERROR PM 5286-- {text} local file name redefined by gengf p=TRUE

Condition Identifier: PME\$LOCAL_NAME_CONFLICT Code: PM 5286

Description: The specified local file name was used more than once on a call to GENERATE_GROUP_FILE with the PERMANENT parameter value of true.

User Action: Check the accuracy of the local file name used on the call to GENGF or use a different file name.

--ERROR PM 5287-- {text} is not a legal date

Condition Identifier: PME\$BAD_DATE Code: PM 5287

Description: Date value is not a legal date.

User Action: See the Statistics Facility Chapter in the NOS/VE System Performance and Maintenance Manual for a description of the date parameter on the DEFINE_GROUP subcommand.

--ERROR PM 5300-- Internal MPE segment overflow.

Condition Identifier: PME\$E_INTERNAL_MPE_SEG_OVERFLOW Code: PM 5300

Description: An internal file has overflowed.

User Action: Inform the site analyst providing appropriate documentation of the error.

--ERROR PM 5301-- Premature end of file encountered on file: {file}.

Condition Identifier: PME\$E_PREMATURE_EOF_ON_FILE Code: PM 5301

Description: The specified file has a format error.

User Action: Regenerate the file.

--ERROR PM 5302-- Error encountered while reading APD_SEQ file: {file}.

Condition Identifier: PME\$E_INTERNAL_APD_READ_ERROR Code: PM 5302

Description: An internal error has occurred on the specified file.

User Action: Inform the site analyst providing appropriate documentation of the error.

--ERROR PM 5303-- Error encountered while intercepting calls - MPE unable to continue.

Condition Identifier: PME\$E_FATAL_INTERCEPT_ERROR Code: PM 5303

Description: An error was encountered while intercepting calls. Measure Program Execution is unable to continue.

User Action: Inform the site analyst providing appropriate documentation of the error.

--ERROR PM 5304-- Target text file name must be included in object file list.

Condition Identifier: PME\$E_TARGET_TEXT_NOT_FILE Code: PM 5304

Description: The target text file name must be included in the object file list.

User Action: Reissue the command with the target text file specified in the FILE parameter value list.

--ERROR PM 5305-- File being read was not created by APD: {file}.

Condition Identifier: PME\$E_FILE_NOT_CREATED_BY_MPE Code: PM 5305

Description: There is a format error on the program measures file.

User Action: Regenerate the program measures file.

--ERROR PM 5306-- A program description has not yet been defined.

Condition Identifier: PME\$E_NO_PROGRAM_DESCRIPTION Code: PM 5306

Description: A SET_PROGRAM_DEFINITION command must be issued before this command.

User Action: Issue a SET_PROGRAM_DEFINITION command and then reissue the current command.

--ERROR PM 5307-- Path name of local files too long for restructure procedure.

Condition Identifier: PME\$PATH_NAME_TOO_LONG_FOR_MPE Code: PM 5307

Description: The path name of the local files is too long for the restructure procedure.

User Action: Shorten the path name.

--ERROR PM 5308-- Unable to complete Measure Program Execution.

Condition Identifier: PME\$E_MPE_LOADER_ABORT Code: PM 5308

Description: The instrumented task load failed.

User Action: Check the load map for errors and correct.

--WARNING PM 5309-- Measure Program Execution environment has been reinitialized.

Condition Identifier: PME\$W_MPE_ENVIRONMENT_RESTORED Code: PM 5309

Description: The Measure Program Execution environment has been reinitialized.

User Action: Reissue SET_PROGRAM_DESCRIPTION and continue.

--ERROR PM 5310-- Missing or empty file on file: {file}.

Condition Identifier: PME\$E_MISSING_OR_EMPTY_FILE Code: PM 5310

Description: A missing or empty file has been specified.

User Action: Issue a command with a valid file.

--ERROR PM 5311-- A connectivity matrix does not exist.

Condition Identifier: PME\$E_NO_CONNECTIVITY_MATRIX Code: PM 5311

Description: A connectivity matrix does not exist.

User Action: Generate a connectivity matrix on the EXECUTE_INSTRUMENTED_TASK command.

--ERROR PM 5312-- Execution time totals do not exist.

Condition Identifier: PME\$E_NO_EXECUTION_TIME_TOTALS Code: PM 5312

Description: Execution time totals do not exist.

User Action: Generate execution time totals by issuing an EXECUTE_INSTRUMENTED_TASK command.

--ERROR PM 5313-- The block name map has not yet been defined.

Condition Identifier: PME\$E_NO_BLOCK_NAME_MAP Code: PM 5313

Description: The block name map has not yet been defined.

User Action: Define a block name map by issuing an EXECUTE_INSTRUMENTED_TASK command.

--ERROR PM 5314-- There are no local program units to print.

Condition Identifier: PME\$E_NO_LOCAL_PROGRAM_UNITS Code: PM 5314

Description: The program being analyzed contains no local units. All units are remote.

User Action: None.

--ERROR PM 5315-- There are no remote program units to print.

Condition Identifier: PME\$E_NO_REMOTE_PROGRAM_UNITS Code: PM 5315

Description: The program being analyzed contains no calls to remote programs.

User Action: None.

--ERROR PM 9900-- Requestor cannot originate the specified system flag.

Condition Identifier: PME\$INVALID_TASK_ORIGIN_FLAG Code: PM 9900

Description: The requester does not have sufficient privilege to issue the specified system flag.

User Action: None.

--ERROR PM 9901-- Requestor cannot originate the specified signal.

Condition Identifier: PME\$INVALID_TASK_ORIGIN_SIGNAL Code: PM 9901

Description: The requester does not have sufficient privilege to issue the specified signal.

User Action: None.

--ERROR PM 9902-- NIL handler on a define handler request

Condition Identifier: PME\$NIL_HANDLER Code: PM 9902

Description: The defined condition handler pointer cannot be NIL.

User Action: Specify a valid handler.

--ERROR PM 9903-- an attempt to replace a handler

Condition Identifier: PME\$HANDLER_ALREADY_DEFINED Code: PM 9903

Description: The condition handler cannot be defined because a condition handler already exists for the specified scope and conditions.

User Action: Check that the correct scope and conditions were specified.

--ERROR PM 9904-- monitor fault, system flag or signal identifier is in error

Condition Identifier: PME\$INVALID_IDENTIFIER Code: PM 9904

Description: Internal system error (monitor fault). The system flag or signal identifier is in error.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--ERROR PM 9905-- requestor does not have adequate privilege to define the handler

Condition Identifier: PME\$DEFINE_PRIVILEGE_ERROR Code: PM 9905

Description: The requester does not have sufficient privilege to define the handler.

User Action: None.

--ERROR PM 9906-- new handler does not have adequate privilege

Condition Identifier: PME\$EXEC_CALL_BRACKET_ERROR Code: PM 9906

Description: The new handler does not have sufficient privilege.

User Action: Change the right brackets of the handler.

--ERROR PM 9907-- requestor specified an nonexistent task

Condition Identifier: PME\$UNKNOWN_RECIPIENT_TASK Code: PM 9907

Description: The specified task does not exist.

User Action: Specify an existing task.

--ERROR PM 9908-- insufficient privilege for request

Condition Identifier: PME\$INSUFFICIENT_PRIVILEGE Code: PM 9908

Description: The request does not have sufficient privilege to be completed.

User Action: None.

--FATAL PM 9909-- system error - monitor fault={text}, buffer full at P={text}, MCR={text}, UCR={text}.

Condition Identifier: PME\$MONITOR_FAULT_BUFFER_FULL Code: PM 9909

Description: The operating system's monitor fault buffer is full.

User Action: Inform the site analyst providing appropriate documentation of the error.

--FATAL PM 9910-- system error - monitor fault={text} with traps disabled at P={text}, DSP={text}, MCR={text}.

Condition Identifier: PME\$FAULT_WITH_TRAPS_DISABLED Code: PM 9910

Description: Internal system error. A fault occurred with traps disabled.

User Action: Inform a site analyst providing appropriate documentation of the error.

--FATAL PM 9911-- system error - invalid DSP={text} at P={text}, MCR={text}, UCR={text}.

Condition Identifier: PME\$INVALID_DYNAMIC_SPACE_PTR Code: PM 9911

Description: Internal system error. An invalid dynamic space pointer was found.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--FATAL PM 9912-- system error - invalid P={text}, DSP={text}, MCR={text}, UCR={text}.

Condition Identifier: PME\$INVALID_P_REGISTER Code: PM 9912

Description: An invalid P register was found (a system error).

User Action: Inform the site analyst providing appropriate documentation of the error.

--FATAL PM 9913-- system error - MCR={text} with traps disabled at P={text}.

Condition Identifier: PME\$MCR_WITH_TRAPS_DISABLED Code: PM 9913

Description: Internal system error. A monitor condition occurred with traps disabled.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--FATAL PM 9914-- system error - unstackable UCR={text} with traps disabled at P={text}.

Condition Identifier: PME\$UCR_WITH_TRAPS_DISABLED Code: PM 9914

Description: Internal system error. An unstackable user condition occurred with traps disabled.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--CATASTROPHIC PM 9915-- system error - please submit job log with PSR.

Condition Identifier: PME\$SYSTEM_ERROR Code: PM 9915

Description: Internal system error.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--CATASTROPHIC PM 9916-- system error - received an undefined broken task fault at P={text}.

Condition Identifier: PME\$UNDEFINED_BROKEN_TASK Code: PM 9916

Description: Internal system error. A broken task fault has occurred.

User Action: Inform a site analyst, providing appropriate documentation of the error.

--ERROR PM 9917-- requestor specified a task recognized as hung by monitor

Condition Identifier: PME\$HUNG_RECIPIENT_TASK Code: PM 9917

Description: The task named as the recipient of a signal from another task is hung and cannot respond.

User Action: Provide the site analyst with appropriate documentation of the error.

--FATAL PM 9920-- broken condition processor - {text}

Condition Identifier: PME\$BROKEN_CONDITION_PROCESSOR Code: PM 9920

Description: A system error has occurred.

User Action: Inform the site analyst of the error providing appropriate documentation of the error.

--FATAL PM 9921-- debug trap with no debugger loaded

Condition Identifier: PME\$INVALID_DEBUG_TRAP Code: PM 9921

Description: Internal system error. A debug trap condition occurred without Debug loaded.

User Action: Inform the site analyst, providing appropriate documentation of the error.

--ERROR PM 9930-- Incorrect millisecond specified: {text}.

Condition Identifier: PME\$INVALID_MILLISECOND Code: PM 9930

Description: An invalid millisecond value has been specified.

User Action: Specify a valid millisecond.

--ERROR PM 9931-- Incorrect second specified: {text}.

Condition Identifier: PME\$INVALID_SECOND Code: PM 9931

Description: An incorrect value was specified.

User Action: Specify a valid second.

--ERROR PM 9932-- Incorrect minute specified: {text}.

Condition Identifier: PME\$INVALID_MINUTE Code: PM 9932

Description: An incorrect minute value was specified.

User Action: Specify a valid minute.

--ERROR PM 9933-- Incorrect hour specified: {text}.

Condition Identifier: PME\$INVALID_HOUR Code: PM 9933

Description: An incorrect hour value was specified.

User Action: Specify a valid hour.

--ERROR PM 9934-- Incorrect month specified: {text}.

Condition Identifier: PME\$INVALID_MONTH Code: PM 9934

Description: An incorrect month value was specified.

User Action: Specify a valid month.

--ERROR PM 9935-- Incorrect day specified: {text}.

Condition Identifier: PME\$INVALID_DAY Code: PM 9935

Description: An incorrect day value was specified.

User Action: Specify a valid day.

--ERROR PM 9936-- Incorrect year specified: {text}.

Condition Identifier: PME\$INVALID_YEAR Code: PM 9936

Description: An incorrect year value was specified.

User Action: Specify a valid year.

--ERROR PM 9937-- Computed value for the year is out of range.

Condition Identifier: PME\$COMPUTED_YEAR_OUT_OF_RANGE Code: PM 9937

Description: The computed value for the year is out of range.

User Action: Recompute the date with a smaller increment.

--ERROR PM 9938-- Arithmetic overflow occurred during computation.

Condition Identifier: PME\$COMPUTE_OVERFLOW Code: PM 9938

Description: An arithmetic overflow occurred during computation.

User Action: Inform a site analyst providing appropriate documentation of the error.

--ERROR PM 9939-- Incorrect time zone specified.

Condition Identifier: PUE\$INVALID_TIME_ZONE Code: PM 9939

Description: Some part of the time zone information is incorrect. It may be outside the allowed range.

User Action: Specify valid hours_from_gmt or minutes_offset.

--ERROR PU 101-- Unexpected reference to file position for parameter {text}.

Condition Identifier: PUE\$UNEXPECTED_FILE_POSITION Code: PU 101

Description: A file position cannot be specified on the indicated parameter.

User Action: Do not specify a file position on the indicated parameter. File position may not be specified on catalog or file parameters.

--ERROR PU 102-- The parameter {text} is too short for this request. {text} is the first value. {text}

Condition Identifier: PUE\$PATH_TOO_SHORT Code: PU 102

Description: The specified parameter is too short for this request. The first value is indicated.

User Action: Replace with the correct path. Catalog paths must contain at least two names, while file paths must contain at least three.

--ERROR PU 103-- Expecting permanent file path, found \$LOCAL.{text} for parameter {text}.

Condition Identifier: PUE\$UNEXPECTED_\$LOCAL Code: PU 103

Description: Only permanent files or permanent file catalogs can be backed up or restored.

User Action: Copy the local file to a permanent file before backup.

--ERROR PU 105-- The NULL_BACKUP_FILE_OPTION parameter is only available when the backup file is of device class NULL.

Condition Identifier: PUE\$NULL_BACKUP_FILE_REQUIRED Code: PU 105

Description: The NULL_BACKUP_FILE_OPTION parameter should only be specified when the backup file is a member of the NULL device class.

User Action: Specify BACKUP_FILE=\$NULL on the call to BACKUP_PERMANENT_FILE when using the NULL_BACKUP_FILE_OPTION of SET_BACKUP_OPTIONS.

--ERROR PU 500-- Unrecoverable error occurred using backup file - {text} {text} {text} {text}.

Condition Identifier: PUE\$UNUSABLE_RESTORE_FILE Code: PU 500

Description: The described error occurred during restore. Recovery was attempted but some data may have been lost.

User Action: Check the list file and the restored file. If using tape, clean the tape unit and retry. Contact the site analyst if problems persist.

--ERROR PU 502-- Unable to find {text} {text} {text} on backup file - no restore performed.

Condition Identifier: PUE\$NO_RESTORE_NO_FIND Code: PU 502

Description: The requested item to restore was not found on the backup file.

User Action: Check the backup list to see what was backed up or use the DISPLAY_BACKUP_FILE subcommand to display what is on the backup file.

--ERROR PU 505-- A cycle was specified on the {text} parameter but not on the FILE parameter.

Condition Identifier: PUE\$UNEXPECTED_CYCLE_ON_NFN Code: PU 505

Description: The NEW_FILE_NAME parameter describes a cycle but the file parameter describes a full file.

User Action: Either specify a cycle on the file parameter or do not specify a cycle on the NEW_FILE_NAME parameter.

--ERROR PU 506-- The selected backup file {text} must be a mass storage or tape file, rather than a {text} file.

Condition Identifier: PUE\$INVALID_DEVICE_TYPE Code: PU 506

Description: It is not permitted to backup to the terminal.

User Action: Backup to mass storage or tape.

--ERROR PU 507-- An unexpected item was found on the backup file . {text} Contact site analyst

Condition Identifier: PUE\$UNEXPECTED_ITEM_REQUESTED Code: PU 507

Description: An unexpected item was found on the specified backup file. The data integrity was lost.

User Action: Display the log, and print the listing from the backup or restore. Contact the site analyst.

--ERROR PU 508-- The new catalog {text} already exists.

Condition Identifier: PUE\$NEW_CATALOG_ALREADY_EXISTS Code: PU 508

Description: RESTORE_CATALOG is attempting to restore an existing catalog.

User Action: A user may: 1.) Use the NEW_CATALOG_NAME on RESTORE_CATALOG to create a new catalog, or 2.) Delete the catalog and try again, or 3.) Use RESTORE_EXISTING_CATALOG.

--ERROR PU 509-- The new file {text} already exists

Condition Identifier: PUE\$NEW_FILE_ALREADY_EXISTS Code: PU 509

Description: RESTORE_FILE is attempting to restore an existing file.

User Action: A user may: 1.) Use the NEW_FILE_NAME on RESTORE_FILE to create a new file, or 2.) Delete the file and try again, or 3.) Use RESTORE_EXISTING_FILE to restore cycles that do not exist.

--ERROR PU 510-- File {text} cycle {text} already exists.

Condition Identifier: PUE\$NEW_CYCLE_ALREADY_EXISTS Code: PU 510

Description: When doing a RESTORE_FILE of a specific cycle, the cycle may already exist.

User Action: Restore the cycle to a non-existing cycle number. \$NEXT may be used to specify the next highest cycle.

--ERROR PU 518-- The requested set {text} does not exist. Contact site analyst.

Condition Identifier: PUE\$UNKNOWN_SET Code: PU 518

Description: An internal error has occurred.

User Action: Use DISPLAY_LOG. Contact the site analyst and submit a programming system report (PSR).

--FATAL PU 519-- The backup file "{text}" is incompatible with the current system (current version: "{text}", recorded version: "{text}"). Contact site analyst

Condition Identifier: PUE\$INCOMPATIBLE_BACKUP_VERSION Code: PU 519

Description: The current system cannot be used to restore the old backup file.

User Action: Contact the site analyst.

--FATAL PU 520-- An unrecoverable write error occurred on the backup file. {text} {text} {text}

Condition Identifier: PUE\$UNRECOVERED_WRITE_ERROR Code: PU 520

Description: An unrecoverable tape error has occurred.

User Action: Retry the backup using a different tape volume and check the list file. Request the operator to clean the tape unit.

--WARNING PU 523-- Unable to read version number from backup file: read: "{text}". Item skipped

Condition Identifier: PUE\$UNABLE_TO_READ_VERSION Code: PU 523

Description: Restore was unable to read the initial data from the backup file. That item was skipped.

User Action: If the restore was from a tape file, check to see if the restore was started on the first volume of the set of backup file tapes. If the restore was started on a volume other than the first volume, that item is skipped. Check the list file to see if the requested item was restored.

--ERROR PU 524-- The {text} parameter is too long for this request.

Condition Identifier: PUE\$PATH_TOO_LONG Code: PU 524

Description: The specified parameter is too long for this request.

User Action: Replace with the correct path. User names may only consist of two names.

--ERROR PU 525-- The {text} {text} {text} is already excluded from backup/delete.

Condition Identifier: PUE\$ITEM_ALREADY_EXCLUDED Code: PU 525

Description: A previous EXCLUDE_CATALOG or EXCLUDE_FILE already excluded the requested item.

User Action: None.

--ERROR PU 527-- Both the FILE parameter and the CATALOG parameter may not be specified on the RESTORE_EXCLUDED_FILE_CYCLES subcommand.

Condition Identifier: PUE\$BOTH_FILE_CATALOG_SPECIFIED Code: PU 527

Description: Either the file or catalog parameter can be specified but not both.

User Action: Specify either FILE or CATALOG, but not both.

--ERROR PU 528-- The NEW_NAME parameter may only be specified if the FILE or CATALOG parameters are specified.

Condition Identifier: PUE\$NEW_NAME_SPECIFIED_ALONE Code: PU 528

Description: If all excluded cycles are to be restored, then the new name parameter should not be specified.

User Action: Do not specify NEW_NAME or specify either FILE or CATALOG.

--ERROR PU 530-- The date portion of the {text} parameter was omitted but is required.

Condition Identifier: PUE\$DATE_PART_OMITTED Code: PU 530

Description: The date portion was omitted on the BEFORE or AFTER parameter of the INCLUDE_CYCLES subcommand.

User Action: Specify a date with the BEFORE or AFTER parameter of the INCLUDE_CYCLES subcommand.

--ERROR PU 531-- The two date_time values are exactly equal. A zero length window of time for INCLUDE_CYCLES is not allowed.

Condition Identifier: PUE\$BEFORE_AFTER_TIME_EQUAL Code: PU 531

Description: The INCLUDE_CYCLES subcommand can be used to specify a window of time. This period of time cannot be zero.

User Action: Specify a different date_time value for one of the parameters on the INCLUDE_CYCLES subcommand.

--ERROR PU 533-- Internal -After a amp\$skip_tape_marks the returned volume position was {text}.

Condition Identifier: PUE\$NOT_AT_TAPEMARK_AFTER_SKIP Code: PU 533

Description: An internal error has occurred.

User Action: Contact the site analyst or submit a programming system report (PSR).

--ERROR PU 534-- You must be the system or family administrator to use the {text} request {text}.

Condition Identifier: PUE\$NOT_SYSTEM_ADMINISTRATOR Code: PU 534

Description: Only the system administrator can use the specified request.

User Action: Users cannot use the specified request.

--ERROR PU 536-- {text} was specified for parameter {text}. {text}{text}{text}{text}{text}{text} may be specified.

Condition Identifier: PUE\$INCORRECT_CYCLE_REFERENCE Code: PU 536

Description: The specified cycle selector is not an allowed value for this parameter.

User Action: Specify one of the allowed selector values displayed in the text.

--WARNING PU 537-- {text} error{text} occurred during subcommand. Check list file {file}. {text}{text}{text}{text}{text}{text}

Condition Identifier: PUE\$ERROR_SUMMARY_STATUS Code: PU 537

Description: Self-explanatory.

User Action: The list file should be checked for the errors. If a single error occurred, it is displayed in the status text.

--ERROR PU 539-- Volume {text} is not active.

Condition Identifier: PUE\$VOLUME_NOT_ACTIVE Code: PU 539

Description: INCLUDE_VOLUME may only be used on configured volumes.

User Action: Specify a volume that is active. Consult a site analyst for a list of volumes.

--INFORMATIVE PU 544-- {text} catalogs, files, or cycles found.

Condition Identifier: PUE\$NUMBER_OF_ITEMS Code: PU 544

Description: While using the DISPLAY_BACKUP_FILE subcommand, more entries were requested than exist on the backup file.

User Action: None.

--ERROR PU 546-- A condition occurred in backup, check list file.

Condition Identifier: PUE\$BACKUP_CONDITION Code: PU 546

Description: A segment access problem occurred during the backup.

User Action: Print the List file to determine the nature of the failure. If the problem persists, contact site analyst.

--ERROR PU 547-- A condition occurred on restore, check list file.

Condition Identifier: PUE\$RESTORE_CONDITION Code: PU 547

Description: A segment access problem occurred during the restore.

User Action: Print the List file to determine the nature of the failure. If the problem persists, contact site analyst.

--ERROR PU 548-- As family administrator you may only use the {text} request on users in your family. The INCLUDE_USERS request may be used to select the users in your family.

Condition Identifier: PUE\$UNOWNED_USERS_INCLUDED Code: PU 548

Description: Certain BACKUP_PERMANENT_FILE commands are available to the family administrator only for users in the administrator's family.

User Action: Use the INCLUDE_USERS subcommand to specify your family.

--ERROR PU 549-- A family catalog may only be restored as a family catalog, on the {text} request.

Condition Identifier: PUE\$MUST_RESTORE_AS_FAMILY Code: PU 549

Description: If a family catalog name is specified on the CATALOG parameter, then only a family catalog may be specified on the NEW_CATALOG_NAME parameter.

User Action: Specify either a master catalog name for the CATALOG parameter or a family catalog name for the NEW_CATALOG_NAME parameter, but not both.

--ERROR PU 550-- A user catalog may not be restored as a family catalog, on the {text} request.

Condition Identifier: PUE\$CANT_RESTORE_AS_FAMILY Code: PU 550

Description: A master catalog name was specified on the CATALOG parameter, but a family catalog name was specified on the NEW_CATALOG_NAME parameter.

User Action: Specify either a family catalog name for the CATALOG parameter or a master catalog name for the NEW_CATALOG_NAME parameter, but not both.

--ERROR PU 551-- A subcatalog may only be restored as a subcatalog, NOT as a FAMILY or USER catalog, on the {text} request.

Condition Identifier: PUE\$MUST_RESTORE_AS_SUBCATALOG Code: PU 551

Description: RESTORE_CATALOG may not be used to restore a subcatalog as either a new family or a new user.

User Action: Specify a subcatalog path for the NEW_CATALOG_NAME parameter.

--ERROR PU 552-- The name specified on the {text} parameter must be different than the name specified on the {text} parameter.

Condition Identifier: PUE\$NEW_NAME_MUST_BE_DIFFERENT Code: PU 552

Description: The values specified for the two parameters must be different.

User Action: Specify a different name for either of the parameters.

--ERROR PU 553-- You must be the operator to use the {text} request.

Condition Identifier: PUE\$NOT_SYSTEM_OPERATOR Code: PU 553

Description: This command can only be executed from the system console.

User Action: No action required.

--ERROR PU 554-- Backup_label_type is UNLABELLED but the tape volume was labelled and the vsn specified on REQUEST_MAGNETIC_TAPE {text} does not match the recorded_vsn on the tape volume.

Condition Identifier: PUE\$LABEL_MISMATCH Code: PU 554

Description: The recorded vsn on the tape volume did not match the VOLUME_SERIAL_NUMBER specified on the REQUEST_MAGNETIC_TAPE command.

User Action: Request the tape with the correct recorded volume serial number.

--WARNING PU 605-- File {text} cycle {text} was not restored because it contains no data, and archive information is not being restored.

Condition Identifier: PUE\$NOTHING_TO_RESTORE Code: PU 605

Description: Specify TRUE for the RESTORE_ARCHIVE_INFORMATION parameter of the SET_RESTORE_OPTIONS subcommand if you need to restore archived file cycles which reside offline.

User Action: Describe the problem on the appropriate PSR.

--ERROR PU 606-- System administrator privilege is required to set the RESTORE_ARCHIVE_INFORMATION parameter of the SET_RESTORE_OPTIONS subcommand to TRUE.

Condition Identifier: PUE\$USER_SET_RAI_TRUE Code: PU 606

Description: Archive information can only be restored by a system administrator.

User Action: Specify RESTORE_ARCHIVE_INFORMATION=FALSE on the SET_RESTORE_OPTIONS subcommand or do the restore from a job which has system administrator privilege.

--ERROR PU 607-- Specifying the cycle number on the BACKUP_FILE subcommand when backing up a released cycle and not including offline data is not permitted, as neither archive information nor data would be backed up.

Condition Identifier: PUE\$EMPTY_CYCLE_PARTITION Code: PU 607

Description: Archive information cannot be included on the backup of a specific file cycle.

User Action: Either specify INCLUDE_ARCHIVE_INFORMATION=FALSE on the SET_BACKUP_OPTIONS subcommand or backup all cycles of the file by omitting the cycle number from the BACKUP_FILE command.

--ERROR PU 608-- Vsns may not be specified with the INCLUDE_VOLUME subcommand when including offline data with the INCLUDE_DATA parameter of the SET_BACKUP_OPTIONS subcommand.

Condition Identifier: PUE\$INCV_AND_OFFLINE_CONFLICT Code: PU 608

Description: Backing up file cycles which reside on a specified mass storage volume is not allowed when OFFLINE_DATA is specified for the INCLUDE_DATA parameter of the SET_BACKUP_OPTIONS subcommand, because file cycles which reside offline are not assigned to mass storage.

User Action: Omit OFFLINE_DATA from the INCLUDE_DATA parameter of the SET_BACKUP_OPTIONS subcommand or omit the INCLUDE_VOLUME subcommand.

--ERROR PU 609-- System administrator privilege is required to set the INCLUDE_ARCHIVE_INFORMATION parameter of the SET_BACKUP_OPTIONS subcommand to TRUE.

Condition Identifier: PUE\$USER_SET_IAI_TRUE Code: PU 609

Description: Only the system administrator is allowed to backup archive information.

User Action: Specify INCLUDE_ARCHIVE_INFORMATION=FALSE on the SET_BACKUP_OPTIONS subcommand or do the backup from a job which has system administrator privileges.

--WARNING PU 610-- Archive information was not restored because a cycle number was specified.

Condition Identifier: PUE\$NO_ARC_INFO_IF_CYCLE_NUMBER Code: PU 610

Description: Archive information cannot be restored when a cycle number is specified on the RESTORE_FILE subcommand.

User Action: Either specify RESTORE_ARCHIVE_INFORMATION=FALSE on the SET_RESTORE_OPTIONS subcommand or restore all cycles of the file by omitting the cycle number from the RESTORE_FILE subcommand.

--ERROR PU 613-- The UPDATE_CYCLE_STATISTICS and RESTORE_ARCHIVE_INFORMATION parameters cannot both be set to TRUE.

Condition Identifier: PUE\$UCS_RAI_CONFLICT Code: PU 613

Description: Archive information cannot be restored if cycle statistics are updated.

User Action: Either specify UPDATE_CYCLE_STATISTICS=FALSE or specify RESTORE_ARCHIVE_INFORMATION=FALSE on the SET_RESTORE_OPTIONS subcommand.

--INFORMATIVE QK 0-- This calculation field name is not within a field.

Condition Identifier: QKE\$CDIN_NO_FIELD Code: QK 0

Description: You placed the calculation field name between fields or after a field.

User Action: Put the name within a field. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 2-- This calculation field name is within more than one field.

Condition Identifier: QKE\$CDIN_MULTIPLE_FIELDS Code: QK 2

Description: The calculation field name in the CALCULATION FIELD IDENTIFIERS window lies under more than one field.

User Action: Delete or insert the necessary characters to align the calculation name with the field. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 4-- This calculation field name contains more than 2 characters.

Condition Identifier: QKE\$CDTEMP_NAME_GT_2_CHAR Code: QK 4

Description: Self-explanatory.

User Action: Reduce the field name to one or two characters. The first character must be alphabetic. If there is a second character, it must be alphanumeric.

--INFORMATIVE QK 6-- A calculation field name beginning with a letter and optionally containing a second alphanumeric character is expected here.

Condition Identifier: QKE\$CDINVALID_TEMP_NAME Code: QK 6

Description: You entered an invalid calculation field name. Examples would be '12' or 'a9o'.

User Action: Correct the name as described in the error message. For more information refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 8-- This calculation field name is not unique.

Condition Identifier: QKE\$CDDUP_TEMP_NAME Code: QK 8

Description: Self-explanatory.

User Action: Change the name of the field so that it is unique. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 12-- You have specified more than one name for this field.

Condition Identifier: QKE\$CDMULTIPLE_TEMP_NAME Code: QK 12

Description: Self-explanatory.

User Action: Delete one calculation field name or move it to another field. For more information, refer to the Quick Online Manual using the index topic 'calculation window'.

--INFORMATIVE QK 14-- You must specify a DESCRIPTION before you may calculate.

Condition Identifier: QKE\$CDNO_DATA_DESCR Code: QK 14

Description: You are trying to CALCULATE before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from another profile through the UTILITY function. For more information, refer to the Quick Online Manual using the index topic 'Description of data' or 'Utility'.

--INFORMATIVE QK 16-- This token is invalid.

Condition Identifier: QKE\$CDINVALID_TOKEN Code: QK 16

Description: You have entered an invalid character or combination of characters in the CALCULATION window. An example would be 'a=%5' or 'a=126.3'.

User Action: Replace the invalid characters with valid characters. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 18-- Data types disagree.

Condition Identifier: QKE\$CDDATA_TYPES_DISAGREE Code: QK 18

Description: In the CALCULATION window, you are including a numeric field and a character field in the same expression; for example: 'n1=c1 where n1 is numeric and c1 is character.

User Action: Remove the expression or correct it so that it only includes one type of data. For more information, refer to the QUICK Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 20-- This expression ended prematurely.

Condition Identifier: QKE\$CDPREMATURE_END Code: QK 20

Description: A formula in your CALCULATION window is not complete. Examples would be 'a=b+' or 'a=(b='.

User Action: Correct the formula. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 22-- A field name is expected here.

Condition Identifier: QKE\$CDEXPECT_FIELD_NAME Code: QK 22

Description: An item in the CALCULATION window does not match any field defined in the CALCULATION FIELD IDENTIFIERS window.

User Action: Replace the incorrect item with an appropriate field name. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 24-- An equals sign is expected here.

Condition Identifier: QKE\$CDEXPECT_EQUALS Code: QK 24

Description: You have omitted an equals sign in the CALCULATION window. For example: 'a b' or 'a b=c'.

User Action: Correct the formula to include the equals sign. For more information, see the Quick Online Manual using the index topic 'formulas'.

--INFORMATIVE QK 26-- Arithmetic calculations on character items are not allowed.

Condition Identifier: QKE\$CDOPERATION_ON_CHAR Code: QK 26

Description: Self-explanatory.

User Action: Remove the calculation or change the calculation to use numeric items. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 28-- Consecutive operands are not allowed.

Condition Identifier: QKE\$CDCONSECUTIVE_OPNDS Code: QK 28

Description: In the CALCULATION window, you entered two consecutive operands; for example: 'a=a(b)' or 'a=6 5'.

User Action: Correct the calculation by inserting an operator such as 'a=a+(b)' or 'a=6-5'. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 30-- Consecutive operators are not allowed.

Condition Identifier: QKE\$CDCONSECUTIVE_OPS Code: QK 30

Description: You have placed two consecutive operators in the CALCULATION window; for example: 'a=b*-c' or 'a=b=/c'.

User Action: Surround embedded unary operators with parentheses; for example: 'a=b*(-c)'. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 32-- An operand is expected here.

Condition Identifier: QKE\$CDEXPECT_OPERAND Code: QK 32

Description: In the CALCULATION window, you entered a formula that is missing an operand, such as 'a=(+)'.
'a=(+)'.

User Action: Insert the appropriate operand such as 'a=(b+c)'. For more information, refer to the QUICK Online Manual using the index topic 'formulas'.

--INFORMATIVE QK 34-- A unary operator (+ or -) is expected here.

Condition Identifier: QKE\$CDEXPECT_UNARY_OP Code: QK 34

Description: Your calculation formula begins with an inappropriate operator; for example: 'a=*bg' or 'b=/ac'

User Action: Correct the formula so that it is in the form result=calculation. For example: 'a=-b*g' or 'b=a/c'. For more information, refer to the Quick Online Manual using the index topic 'formulas.'

--INFORMATIVE QK 36-- This is an unmatched parenthesis.

Condition Identifier: QKE\$CDUNMATCHED_PAREN Code: QK 36

Description: Self-explanatory.

User Action: Correct the formula, inserting or deleting parentheses as needed. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 38-- An internal error (no fault of user) has occurred.

Condition Identifier: QKE\$CDINTERNAL_ERROR Code: QK 38

Description: The indicated system error was detected during processing of your calculation formulas. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 40-- You must specify an IN file and DESCRIPTION before you may calculate.

Condition Identifier: QKE\$CD_NO_IN_FILE Code: QK 40

Description: You are trying to specify calculation formulas before specifying the INPUT file and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data', or 'Utility'.

--INFORMATIVE QK 112-- An internal bad parameter error occurred while processing this field.

Condition Identifier: QKE\$COBAD_PARAMETERS Code: QK 112

Description: The indicated system error was detected during processing of a field. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 116-- Division by zero is not allowed.

Condition Identifier: QKE\$CODIVISION_BY_ZERO Code: QK 116

Description: Self-explanatory.

User Action: Modify the data or the CALCULATION formula so that you don't divide by zero. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 118-- Invalid data for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCINVALID_BDP_DATA Code: QK 118

Description: Your sequential input file contains non-numeric characters in a field described as numeric.

User Action: Modify the description; use a different input file or use the Quick editor to correct the field. For more information, refer to the Quick Online Manual using the index topic 'description of data'.

--INFORMATIVE QK 122-- Overflow for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCOVERFLOW Code: QK 122

Description: There was an overflow in your sequential file at the position indicated in the error message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 124-- Underflow for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCUNDERFLOW Code: QK 124

Description: There was an underflow on your sequential edit file in the location indicated in the error message.

User Action: Use the Quick editor to correct the invalid data.

--INFORMATIVE QK 126-- Indefinite value for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCINDEFINITE Code: QK 126

Description: There is an indefinite value in your sequential file at the position indicated in this message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 128-- Infinite value for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCINFINITE Code: QK 128

Description: There is an infinite value in your sequential file at the point indicated in the error message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 130-- Bad parameter internal error for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCBAD_PARAMETERS Code: QK 130

Description: You have encountered an error within Quick.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 134-- Division by zero for field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LOCDIVISION_BY_ZERO Code: QK 134

Description: Self-explanatory.

User Action: Modify the data or the CALCULATION formula so that you don't divide by zero. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 136-- Invalid data for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISINVALID_BDP_DATA Code: QK 136

Description: Your indexed sequential file contains non-numeric characters in a field described as numeric.

User Action: Modify the description; use a different input field or use the Quick editor to correct the field. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 138-- Loss of significant digits for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISLOSS_OF_SIGNIFICANCE Code: QK 138

Description: In your indexed sequential file, you are trying to move or calculate a value which does not fit in the field you described. For example, you could be trying to store 1234 in a field described as 999.

User Action: Examine the error message to determine which field is too short. Either increase the field length or modify the calculation that caused this error. For more information, refer to the Quick Online Manual using the index topics 'Description of data' and 'Calculation window'.

--INFORMATIVE QK 140-- Overflow for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISOVERFLOW Code: QK 140

Description: There was an overflow in your indexed sequential file at the position indicated in the error message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 142-- Underflow for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISUNDERFLOW Code: QK 142

Description: There was an underflow on your indexed sequential file in the location indicated in the message.

User Action: Use the Quick editor to correct the invalid data.

--INFORMATIVE QK 144-- Indefinite value for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISINDEFINITE Code: QK 144

Description: There is an indefinite value in your indexed sequential file at the position indicated in this message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 146-- Infinite value for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISINFINITE Code: QK 146

Description: There is an infinite value in your indexed sequential file at the position indicated in the error message.

User Action: Use the Quick editor to correct the error.

--INFORMATIVE QK 148-- Bad parameter internal error for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISBAD_PARAMETERS Code: QK 148

Description: You have encountered an error within Quick.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 151-- Division by zero for field {text}, key {text}, file {file}.

Condition Identifier: QKE\$ISDIVISION_BY_ZERO Code: QK 151

Description: Self-explanatory.

User Action: Modify the data or the CALCULATION formula so that you don't divide by zero. For more information, refer to the Quick Online Manual using the index topic 'Calculation window'.

--INFORMATIVE QK 200-- A column which is blank in both rows of this window in order to separate the fields is expected here.

Condition Identifier: QKE\$DIMPROPER_BLANK_SEPARATION Code: QK 200

Description: In the DESCRIPTION OF DATA, you have characters in the second line that do not line up with characters on the first line. Be aware that after Quick inserts a default output description (for example 99.9 for an output description of 99v9), you must either delete the output description or keep it aligned when you modify or move the input description.

User Action: Insert or delete characters so that the descriptions on the second line line up with the fields on the first line. For more information, refer to the Quick Online Manual using the index topic 'description of data'.

--INFORMATIVE QK 202-- This character may not appear in the description.

Condition Identifier: QKE\$DINVALID_CHAR Code: QK 202

Description: The DESCRIPTION OF DATA may contain the following characters: Xx Ii Ff Vv (). 0 1 2 3 4 5 6 7 8 9.

User Action: Delete or replace the incorrect characters. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 204-- An input description containing 9's may contain V to indicate an assumed decimal point, but it may not contain a decimal point.

Condition Identifier: QKE\$DIN_PT_INVALID Code: QK 204

Description: Self-explanatory.

User Action: Remove the decimal point. For further information refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 206-- A field described as F requires an output description containing 9's and an optional decimal point.

Condition Identifier: QKESDF_REQ_OUT_PIC Code: QK 206

Description: Self-explanatory.

User Action: Enter an output description containing 9s and an optional decimal point, or re-describe the field. For more information, refer to the Quick Online Manual using the index topic 'F'.

--INFORMATIVE QK 208-- A field described as I requires an output description containing 9's or a date output description.

Condition Identifier: QKESDI_REQ_OUT_PIC Code: QK 208

Description: Self-explanatory.

User Action: Enter an output description containing 9s or re-describe the field. For more information, refer to the Quick Online Manual using the index topic 'I'.

--INFORMATIVE QK 212-- F is not allowed in the output description; you need a description containing 9's and an optional decimal point.

Condition Identifier: QKESDF_INVALID_OUT_PIC Code: QK 212

Description: You have an F on the second line of the DESCRIPTION OF DATA. F is allowed only on the first line.

User Action: Use a combination of 9s and a decimal point instead of F. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 214-- I is not allowed in the output description; you need a description containing 9's.

Condition Identifier: QKESDI_INVALID_OUT_PIC Code: QK 214

Description: You have an I on the second line of the DESCRIPTIONS OF DATA window. I is allowed only on the first line.

User Action: Use 9s instead of I. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 216-- Input and output descriptions disagree.

Condition Identifier: QKESDIN_OUT_DISAGREE Code: QK 216

Description: The input (first line) and output (second line) of the DESCRIPTION OF DATA disagree; one describes character data and the other describes numeric data.

User Action: Change the input or output description so that one agrees with the other. For more information, refer to the Quick Online Manual using the index topic 'description of data window.'

--INFORMATIVE QK 218-- This character is illegal in this context.

Condition Identifier: QKESDINVALID_CONTEXT Code: QK 218

Description: This character can be used in the DESCRIPTION OF DATA but not in this context.

User Action: Delete or replace the incorrect characters. For information on proper context, refer to the Quick Online Manual. As an example, if you are entering the DESCRIPTION OF DATA, use the index topic 'Description of data'.

--INFORMATIVE QK 220-- A numeric description may contain up to 18 9's or z's.

Condition Identifier: QKESDNUMERIC_GR_18 Code: QK 220

Description: In the DESCRIPTION OF DATA window, you have used more than 18 characters to describe a numeric field. For example: 9(20) or 9(17)v99.

User Action: Shorten the field to 18 characters 9(18) or 9(16)v99. For more information, refer to the Quick Online Manual using the index topic 'Description of data window.'

--INFORMATIVE QK 222-- A numeric description must contain at least 1 numeric character.

Condition Identifier: QKE\$DNUMERIC_LT_1 Code: QK 222

Description: You have entered a numeric description with no numeric characters; for example: '.' or 'v'.

User Action: You must include at least one 9. Delete or replace the incorrect description. For more information, refer to the Quick Online Manual using the index topic 'Description of data window.'

--INFORMATIVE QK 224-- An integer description may contain up to 8 characters.

Condition Identifier: QKE\$DI_GR_8 Code: QK 224

Description: In the DESCRIPTION OF DATA window, you have used more than eight characters to describe an integer field; for example: IIIIIIII or I(9).

User Action: Shorten the field to eight characters or less such as IIIIIIII or I(8). For more information refer to the Quick Online Manual using the index topic 'description of data window'.

--INFORMATIVE QK 226-- A hardware error (no fault of user) occurred during picture analysis.

Condition Identifier: QKE\$DHARDWARE_ERROR Code: QK 226

Description: The indicated system error was detected during processing of your description. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 228-- An edit error (no fault of user) occurred during picture analysis.

Condition Identifier: QKE\$DEDIT_ERROR Code: QK 228

Description: The indicated system error was detected during processing of your description. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 230-- An internal error (no fault of user) occurred with a floating picture.

Condition Identifier: QKE\$DF_ERROR Code: QK 230

Description: The indicated system error was detected during processing of your description. This error indicates a malfunction of the system.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 232-- A field may contain up to 15000 characters.

Condition Identifier: QKE\$DITEM_GR_MAX Code: QK 232

Description: You have entered more than 800 characters for one field. As an example: x(809)

User Action: Reduce the field size to 800 or less. For more information, refer to the Quick Online Manual using the index topic 'Description of data.'

--INFORMATIVE QK 234-- The right parenthesis is missing.

Condition Identifier: QKE\$DRIGHT_PAREN_MISSING Code: QK 234

Description: Your 'Description of data' window is missing the right parenthesis. An example would be X(5.

User Action: Insert the right parenthesis. In the example, this would be X(5). For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 236-- The right parenthesis is misplaced.

Condition Identifier: QKE\$DRIGHT_PAREN_MISPLACED Code: QK 236

Description: You have used a right parenthesis incorrectly. Examples would be ')' or ')12'

User Action: Delete or correct the parentheses; an example would be '9(4)'. For more information refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 238-- The left parenthesis is misplaced.

Condition Identifier: QKE\$DLEFT_PAREN_MISPLACED Code: QK 238

Description: You used a left parenthesis incorrectly. An example would be 'C'.

User Action: Delete or correct the parenthesis; for example, '9(4)'. For more information refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 240-- The repetition count is invalid or misplaced.

Condition Identifier: QKE\$DBAD_REPETITION_COUNT Code: QK 240

Description: You have used a number incorrectly in the DESCRIPTION OF DATA. An example would be 9(04 or 4.

User Action: Correct the number, inserting parentheses where needed, such as 9(4). For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 248-- A column which is blank in all rows of the COLUMN HEADINGS window in order to separate the fields is expected here.

Condition Identifier: QKE\$DSEPARATOR_NO_POS Code: QK 248

Description: There is no column which is blank in all rows of the COLUMN HEADINGS window which can be used to separate the fields.

User Action: Adjust your column headings to leave a blank column in all rows between each field. For more information, refer to the Quick Online Manual using the index topic 'Field separation'.

--INFORMATIVE QK 252-- You need {text} more blanks after this description in order for Quick to expand your description.

Condition Identifier: QKE\$DMORE_BLANKS_AFTER Code: QK 252

Description: When using parentheses in a data field description, the description is shorter than the actual field length. As an example, x(6) uses four characters to describe the six character field xxxxxx. You have not allowed enough spaces for the description to be expanded.

User Action: You must insert blanks after the right parenthesis. Insert the number of blanks specified in the error message. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 254-- This field is {text} columns too long to fit in the 800 column description.

Condition Identifier: QKE\$DMORE_BLANKS_800 Code: QK 254

Description: The last field in your DESCRIPTION OF DATA makes the description exceed the 800 column limit.

User Action: Shorten or eliminate fields so that your description does not exceed 800 characters. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 256-- A described field must begin at character {text} to begin at the same character as the key.

Condition Identifier: QKE\$DKEY_STARTS_AT_FIELD Code: QK 256

Description: Your indexed sequential file has a key which begins at the character specified in the error message. You have not described a field which starts at that character.

User Action: Change your description to describe a field which starts at the character specified in the error message. For more information, refer to the Quick Online Manual using the index topic 'Description of data window'.

--INFORMATIVE QK 258-- This field must be described as I({text}) to match the file's integer key.

Condition Identifier: QKE\$DINTEGER_KEY_PICTURE Code: QK 258

Description: You are describing an indexed sequential file which has an integer key containing no characters where n is specified in the error message. You have described a field which starts at the same position as the key, but is described differently than the key.

User Action: Describe the field as specified in the error message.

--INFORMATIVE QK 260-- Your description (currently {text} characters) must be longer than the file's nonembedded key ({text} characters).

Condition Identifier: QKESDREC_LESS_EXKEY Code: QK 260

Description: Your indexed sequential file has a non-embedded key length. The first part of your description should describe the non-embedded key and the rest should describe the record. You have not included enough characters to describe the record.

User Action: Change your description to describe the non-embedded key and the record. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 262-- First specify an input file, then press {text}.

Condition Identifier: QKESDNEED_IN_FILE Code: QK 262

Description: You are pressing DESCR before you have specified the input file.

User Action: Press IN and specify the input file. Then you can use DESCR. For more information, refer to the Quick Online Manual using the index topic 'Input file'.

--INFORMATIVE QK 264-- Your description (currently {text} characters) may not exceed the file's MAXIMUM_RECORD_LENGTH ({text} characters).

Condition Identifier: QKESDLENGTH_EXCEEDS_MAXRL Code: QK 264

Description: An indexed sequential file has a maximum record length. Your description describes more characters than this maximum.

User Action: Change your description to describe only this maximum. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 266-- This F data item which is word aligned and occupies 8 characters must either be all within or all beyond the {text} char nonembedded key.

Condition Identifier: QKESDFLOAT_IN_OR_OUT Code: QK 266

Description: You are describing an indexed sequential file which has a non-embedded key which contains n characters, where n is specified in the error message. The first n characters of your description describes the non-embedded key, and the rest of your description describes the record. You have described an F item such that part of it is in the non-embedded key and part of it is in the record.

User Action: Correct your description so that the F item is either completely within the non-embedded key, or else completely within the record.

--INFORMATIVE QK 268-- Your description (currently {text} characters) may not exceed the file's KEY_LENGTH ({text} characters) plus its MAXIMUM_RECORD_LENGTH ({text} characters).

Condition Identifier: QKESDLEN_EXCEEDS_MAXRL_EXKEY Code: QK 268

Description: Your indexed sequential file has a maximum record length and a non-embedded key length. Your description describes more characters than the sum of these two lengths.

User Action: Change your description to describe only the sum of these two lengths. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 300-- The columns between fields must be blank.

Condition Identifier: QKESFIELD_MUST_BE_BLANK Code: QK 300

Description: You are trying to put data between the fields.

User Action: Delete the characters between the fields. When editing your file, tab or position the cursor to the beginning of each field. For more information, refer to the Quick Online Manual using the index topic 'Edit'.

--INFORMATIVE QK 302-- This field contains too many digits to the left of the decimal point.

Condition Identifier: QK\$ETOO_MANY_DIGITS_LEFT_PT Code: QK 302

Description: While editing y; our file, you entered a number that had more digits to the left of the decimal than was in the numeric description. For example, the description was '999v99' and you entered '1234.5'.

User Action: Reduce the number (for example: 123.45) or increase the description (for example: 9999v99). For more information, refer to the Quick Online Manual using the index topics 'Description of data' and 'Edit'.

--INFORMATIVE QK 304-- This field contains too many digits to the right of the decimal point.

Condition Identifier: QK\$ETOO_MANY_DIGITS_RIGHT_PT Code: QK 304

Description: While editing your file, you entered a number that had more digits to the right of the decimal than was specified in the numeric description. For example, the description was '999v99' and you entered '12.345'.

User Action: Reduce the number of digits (for example, 12.34) or increase the description (for example, 999v999). For more information, refer refer to the Quick Online Manual using the index topics 'Description of data' and 'edit'.

--INFORMATIVE QK 306-- This field may not contain a decimal point.

Condition Identifier: QK\$EINVALID_POINT Code: QK 306

Description: Self-explanatory.

User Action: Remove the decimal point or redefine the field to include a decimal point. For more information, refer to the Quick Online Manual using the index topic 'Edit' or 'Description of data window'.

--INFORMATIVE QK 308-- You already have a record with this key value.

Condition Identifier: QK\$EDUP_PRIMARY_KEY Code: QK 308

Description: When editing your indexed sequential file, you are attempting to duplicate an existing primary key while attempting to insert or update a record.

User Action: Use a different value in the key field.

--INFORMATIVE QK 310-- You have not given a SELECT condition.

Condition Identifier: QK\$ENO_SELECT Code: QK 310

Description: While editing your file, you are pressing SHOW N or SHOW P when you have not entered a SELECT condition.

User Action: To use SHOW N or SHOW P, enter a select condition. For more information, refer to the Quick Online Manual using the index topic 'Select'.

--INFORMATIVE QK 312-- No record satisfied your SELECT condition.

Condition Identifier: QK\$ENO_RECORD_SATISFY Code: QK 312

Description: While in EDIT, you pressed SELECT and entered a select condition, then SHOW N or SHOW P. No record in your file satisfied the condition you entered. For example, you could have entered '>98' on a 999 field, then pressed SHOW N. There were no records with this field value above 98, so this error message was issued.

User Action: No action necessary.

--INFORMATIVE QK 314-- Put the cursor on the record to delete, then press DELETE.

Condition Identifier: QK\$ECURSOR_DELETE Code: QK 314

Description: You are pressing DELETE when the cursor is not on any row in the EDIT FILE window.

User Action: Put the cursor on the row you want to delete, then press the DELETE function key. For more information, refer to the Quick Online Manual using the index topic 'delete'.

--INFORMATIVE QK 316-- Specify data description, then EDIT.

Condition Identifier: QKE\$EPRESS_DESCR Code: QK 316

Description: You are pressing EDIT before you have entered the description of data.

User Action: Press DESCR and enter the description. For more information, refer to the Quick Online Manual using the index topic 'description of data window'.

--INFORMATIVE QK 318-- You cannot edit a file which is assigned to your terminal.

Condition Identifier: QKE\$ECANT_EDIT_THIS_FILE Code: QK 318

Description: You assigned your input file to your terminal. You cannot edit this file.

User Action: Specify an input file which is not assigned to your terminal.

--INFORMATIVE QK 319-- You cannot edit a NOS/VE standard file.

Condition Identifier: QKE\$ECANT_EDIT_NULL_FILE Code: QK 319

Description: You specified \$NULL or a NOS/VE standard file such as \$INPUT as your INPUT FILE NAME.

User Action: If you want to edit the file, specify another file name.

--INFORMATIVE QK 320-- Your file has insufficient access permission for you to save any changes. You may make changes within the workspace, but you may not save these changes.

Condition Identifier: QKE\$EINSUFFICIENT_PERMISSION Code: QK 320

Description: In order to SAVE your changes, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to SAVE your changes, get these permissions.

--INFORMATIVE QK 321-- Your file has insufficient access permission for you to save any changes. You must abandon your changes.

Condition Identifier: QKE\$EINSUFF_PERM_ABANDON Code: QK 321

Description: In order to save changes to your file, you must have READ, MODIFY, APPEND, and SHORTEN permissions.

User Action: You must abandon your changes. If you need to save changes in the future, get these permissions.

--INFORMATIVE QK 322-- You may only edit sequential and indexed-sequential files with this version of Quick.

Condition Identifier: QKE\$EONLY_EDIT_SEQ_IS Code: QK 322

Description: You are trying to edit a file which is not sequential or indexed sequential.

User Action: You cannot edit this type of file with Quick. Use a sequential or indexed sequential file instead.

--INFORMATIVE QK 324-- Put the cursor where the record will be inserted, then press INSERT.

Condition Identifier: QKE\$ECURSOR_INSERT Code: QK 324

Description: You are pressing INSERT when the cursor is not in the EDIT FILE window.

User Action: Put the cursor on or after an existing record, then press the INSERT function key. For more information, refer to the Quick Online Manual using the index topic 'insert'.

--INFORMATIVE QK 326-- Put the cursor on or after an existing row, then press INSERT.

Condition Identifier: QKE\$ECURSOR_INSERT_EXIST Code: QK 326

Description: You are pressing INSERT when the cursor is more than one line past the last row.

User Action: Put the cursor on or after an existing row, then press the INSERT function key. For more information, refer to the Quick Online Manual using the index toipic 'insert'.

--INFORMATIVE QK 328-- Put the cursor on the record to update, then press UPDATE.

Condition Identifier: QKE\$ECURSOR_UPDATE Code: QK 328

Description: You are pressing UPDATE when the cursor is not on any row in the EDIT FILE window.

User Action: Put the cursor on the row you wish to update, then press the UPDATE function key. For more information, refer to the Quick Online Manual using the index topic 'update'.

--INFORMATIVE QK 330-- You made no changes to your file.

Condition Identifier: QKE\$ENO_CHANGES Code: QK 330

Description: You went into the EDIT mode but did not change your file.

User Action: No action is necessary.

--INFORMATIVE QK 332-- You have not given a SELECT condition. Press SELECT to do so.

Condition Identifier: QKE\$ENO_SELECT_PRESS_SELECT Code: QK 332

Description: While editing your file, you are pressing SHOW N or SHOW P when you have not entered a SELECT CONDITION. For more information, refer to the Quick Online Manual using the index topic 'Select'.

User Action: Not applicable.

--INFORMATIVE QK 334-- Your file has insufficient access permission for you to delete.

Condition Identifier: QKE\$ENO_DELETE_PERMISSION Code: QK 334

Description: In order to DELETE records, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to delete records, get these permissions.

--INFORMATIVE QK 336-- Your description (currently {text} characters) must be expanded to include the file's MINIMUM_RECORD_LENGTH ({text} characters).

Condition Identifier: QKE\$EMIN_LENGTH Code: QK 336

Description: Your indexed sequential file has a minimum record length. Your description describes fewer characters than this minimum.

User Action: Change your description to describe the minimum record length. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 338-- Your description (currently {text} chars) must be expanded to include the file's KEY_LENGTH ({text} chars) plus its MINIMUM_RECORD_LENGTH ({text} chars).

Condition Identifier: QKE\$EMIN_LENGTH_EXKEY Code: QK 338

Description: Your indexed sequential file has a non-embedded key length and a minimum record length. You described fewer characters than the sum of these two lengths.

User Action: Change your description to describe the sum of these two lengths. For more information, refer to the Quick Online Manual using the index topic 'Description of data'.

--INFORMATIVE QK 340-- Quick only supports uncollated and integer key types.

Condition Identifier: QKE\$EONLY_UNCOL_INTEGER_KEY Code: QK 340

Description: You are attempting to edit an indexed sequential file whose key type is neither uncollated nor integer.

User Action: You may not edit this file. Use a sequential file or an indexed sequential file with uncollated or integer key.

--INFORMATIVE QK 342-- Your file has insufficient access permission for you to make any changes.

Condition Identifier: QK\$EIS_INSUFFICIENT_PERM Code: QK 342

Description: In order to make any changes to your file, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to make changes, get these permissions.

--INFORMATIVE QK 344-- Your file has insufficient access permission for you to insert.

Condition Identifier: QK\$ENO_INSERT_PERMISSION Code: QK 344

Description: In order to insert, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to insert, get these permissions.

--INFORMATIVE QK 346-- Your file has insufficient access permission for you to update.

Condition Identifier: QK\$ENO_UPDATE_PERMISSION Code: QK 346

Description: In order to UPDATE, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to UPDATE, get these permissions.

--INFORMATIVE QK 348-- Your file has insufficient access permission for you to undo.

Condition Identifier: QK\$ENO_UNDO_PERMISSION Code: QK 348

Description: In order to UNDO, you must have READ, APPEND, MODIFY, and SHORTEN permissions.

User Action: If you need to UNDO, get these permissions.

--INFORMATIVE QK 350-- This record duplicates an existing alternate key value.

Condition Identifier: QK\$EDUP_ALTERNATE_KEY Code: QK 350

Description: While editing your indexed sequential file, you attempted to insert or update a record such that your attempt would duplicate an existing alternate key.

User Action: Enter a key value that is not a duplicate value.

--INFORMATIVE QK 352-- You must specify an IN file and a DESCRIPTION before you may edit.

Condition Identifier: QK\$EPRESS_IN_DESCR Code: QK 352

Description: You are trying to EDIT a file before specifying the input file and description.

User Action: Specify the input file and then the description. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data'.

--INFORMATIVE QK 358-- There are no changes to UNDO.

Condition Identifier: QK\$E_NO_CHANGES_TO_UNDO Code: QK 358

Description: You are pressing UNDO when you haven't made any changes to undo.

User Action: No action necessary.

--INFORMATIVE QK 360-- FILE {file} is empty.

Condition Identifier: QK\$E_FILE_EMPTY Code: QK 360

Description: You pressed WHERE while editing an empty file.

User Action: No action is necessary.

--INFORMATIVE QK 362-- Due to an internal error, file position cannot be determined.

Condition Identifier: QK\$E_LINE_NOT_FOUND Code: QK 362

Description: You have encountered an error within Quick.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 366-- Decimal number contains an invalid non-numeric character.

Condition Identifier: QK\$E_INVALID_DIGIT Code: QK 366

Description: Self-explanatory.

User Action: Delete the non-numeric character or replace it with a numeral. For more information, refer to the Quick Online Manual using the index topic 'edit'.

--INFORMATIVE QK 400-- You must specify a description before you may execute.

Condition Identifier: QK\$EXNEED_DESCRIPTION Code: QK 400

Description: You are trying to XECUTE before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from another profile through the UTILITY function. For more information, refer to the Quick Online Manual using the index topic 'Description of data' or 'Utility'.

--INFORMATIVE QK 402-- You cannot execute using a file which is assigned to your terminal.

Condition Identifier: QK\$EXCANT_EXECUTE_FILE Code: QK 402

Description: You assigned your input file to your terminal. You cannot execute with this file.

User Action: Specify and input file which is not assigned to your terminal.

--INFORMATIVE QK 403-- You cannot execute using a NOS/VE standard file.

Condition Identifier: QK\$EXCANT_READ_NULL_FILE Code: QK 403

Description: You specified \$NULL or a NOS/VE standard file such as \$INPUT as your INPUT FILE NAME.

User Action: Specify an input file which is neither \$NULL nor a NOS/VE standard file.

--INFORMATIVE QK 404-- You must specify an IN file and a DESCRIPTION before you may execute.

Condition Identifier: QK\$EXNEED_IN_DESCR Code: QK 404

Description: You are trying to XECUTE before specifying the INPUT file and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data' or 'Utility'.

--INFORMATIVE QK 412-- {text} records written to {file}.

Condition Identifier: QK\$EXLINES_WRITTEN Code: QK 412

Description: You used the XECUTE function and your output is written to a file. This message tells you how many records were placed in the file.

User Action: No action necessary.

--INFORMATIVE QK 413-- {text} tuples, {text} records written to {file}.

Condition Identifier: QK\$EXTUPLES_WRITTEN Code: QK 413

Description: You used the OUT function to specify that your output is written to a DIF file and used the XECUTE function to write the file. This message tells you how many DIF tuples and how many records were placed in the file.

User Action: No action necessary.

--INFORMATIVE QK 414-- Errors written to {file}.

Condition Identifier: QK\$EXERROR_FILE Code: QK 414

Description: Quick detected one or more errors while you were using VIEW, XECUTE, or EDIT. These errors were written to the error message file specified when executing the QUICK command (default is QUICK_ERRORS).

User Action: View the error file with an editor and take action according to the errors on the error file.

--INFORMATIVE QK 420-- You cannot write to a file which is assigned to your terminal.

Condition Identifier: QK\$EXOUTPUT_ASSIGNED_TERMINAL Code: QK 420

Description: You assigned your output file to your terminal. You cannot write to this file.

User Action: If you want to write to the terminal, use the Quick output option 'Display at terminal'.

--INFORMATIVE QK 421-- You cannot write to a NOS/VE standard file.

Condition Identifier: QK\$EXOUTPUT_TO_NULL Code: QK 421

Description: You specified \$NULL or a NOS/VE standard file such as \$INPUT as your OUTPUT FILE NAME.

User Action: If you want to write to your file, specify an output file which is neither \$NULL nor a NOS/VE standard file.

--INFORMATIVE QK 500-- This field name is not within a field.

Condition Identifier: QK\$FIN_NO_FIELD Code: QK 500

Description: The name in the FIELD IDENTIFIERS window does not line up with any field.

User Action: Correct the field name. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 505-- This field name is within more than one field.

Condition Identifier: QK\$FIN_MULTIPLE_FIELDS Code: QK 505

Description: The name in the FIELD IDENTIFIERS window lies under more than one field.

User Action: Delete or insert the necessary characters to align the field identifier. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 510-- You may not exclude a field for which statistics are being accumulated.

Condition Identifier: QK\$FEXCLUDE_TOTAL Code: QK 510

Description: You have specified that a field be totaled, then excluded it from the ORDER OF FIELDS.

User Action: Add the name to the ORDER OF FIELDS or delete it from the FIELD IDENTIFIERS. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 520-- You must include at least one field.

Condition Identifier: QK\$FALL_EXCLUDED Code: QK 520

Description: In the FIELD IDENTIFIERS window, you have identified fields but you have not specified any in the ORDER OF FIELDS window.

User Action: Specify at least one field, or delete all identifiers from the FIELD IDENTIFIERS window. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 525-- You must specify a DESCRIPTION before you may specify fields.

Condition Identifier: QK\$FNO_DATA_DESCR Code: QK 525

Description: You are trying to specify FIELDS before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from another profile through the UTILITY function. For more information, refer to the Quick Online Manual using the index topic 'Description of data' of 'Utility'.

--INFORMATIVE QK 530-- This field name contains more than 2 characters.

Condition Identifier: QK\$FTEMP_NAME_GT_2_CHAR Code: QK 530

Description: Self-explanatory.

User Action: Reduce the field name to one or two characters. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 535-- A field name beginning with a letter and optionally containing a second alphanumeric character is expected here.

Condition Identifier: QK\$FINVALID_TEMP_NAME Code: QK 535

Description: You have entered an invalid field name, for example: 'a\$' or '1a'.

User Action: Correct the field name. For more information refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 540-- You have specified more than one field name for this field.

Condition Identifier: QK\$FMULTIPLE_TEMP_NAME Code: QK 540

Description: Self-explanatory.

User Action: Delete one field name or move it to another field. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 545-- This is the second time this field name is included.

Condition Identifier: QK\$FFIELD_MULTIPLE_TIMES Code: QK 545

Description: Self-explanatory.

User Action: Remove one occurrence of the field name. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 550-- You did not specify this name in the FIELD IDENTIFIERS window.

Condition Identifier: QK\$FUNKNOWN_NAME Code: QK 550

Description: Self-explanatory.

User Action: Specify the name or replace it with a name that you already specified. For more information, refer to the Quick Online Manual using the index topic 'Fields'.

--INFORMATIVE QK 555-- You must specify an IN file and DESCRIPTION before you may specify fields.

Condition Identifier: QK\$F_NO_IN_FILE Code: QK 555

Description: You are trying to specify FIELDS before specifying the INPUT file and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data', or 'Utility'.

--ERROR QK 600-- Your FROM file may not be a NOS/VE standard file.

Condition Identifier: QK\$F_FROM_FILE_NULL_DEVICE Code: QK 600

Description: While running Quick in execute-only mode, you specified a FROM file that is \$NULL or a NOS/VE standard file such as \$INPUT.

User Action: Execute the QUICK command again, specifying a FROM file that is neither \$NULL nor a NOS/VE standard file.

--ERROR QK 601-- Your FROM file may not be assigned to your terminal.

Condition Identifier: QK\$F_FROM_FILE_ASSIGN_TERMINAL Code: QK 601

Description: While running Quick in execute-only mode, you specified a FROM file that is assigned to your terminal.

User Action: Execute the QUICK command again, specifying a FROM file that is not assigned to your terminal.

--ERROR QK 602-- Your LISTABLE_PROFILE may not be a NOS/VE standard file.

Condition Identifier: QKE\$F_LIST_PROFILE_NULL_DEVICE Code: QK 602

Description: The file that you specified for your listable profile is a NOS/VE standard file.

User Action: Specify a file that is not a NOS/VE standard file.

--ERROR QK 603-- Your LISTABLE_PROFILE may not be assigned to your terminal.

Condition Identifier: QKE\$F_LIST_PROFILE_ASSIGN_TERM Code: QK 603

Description: The file that you specified for your listable profile is assigned to your terminal.

User Action: Specify a file that is not assigned to your terminal.

--INFORMATIVE QK 610-- You may only choose one output option.

Condition Identifier: QKE\$F_DUPLICATE_OPTION Code: QK 610

Description: You have placed an 'x' by more than one option in the OUTPUT OPTIONS window.

User Action: Remove all but one x by entering spaces. For more information, refer to the Quick Online Manual using the index topic 'Specifying output destination'.

--INFORMATIVE QK 615-- A file name is required for this output option.

Condition Identifier: QKE\$F_FILE_NAME_REQUIRED Code: QK 615

Description: You have specified one of the 'write' options in the OUTPUT OPTIONS window, but have omitted the file name in the OUTPUT FILENAME, IF ANY window.

User Action: Enter a file name in the OUTPUT FILENAME, IF ANY window, or change the output option to 'Display at terminal.'

--INFORMATIVE QK 700-- The preceding error occurred at field {text}, record {text}, file {file}.

Condition Identifier: QKE\$LSEQUENTIAL Code: QK 700

Description: There was an error in your file at the location indicated in this message.

User Action: Refer to the preceding error message to determine the cause of the error. Use the Quick editor to correct the error.

--INFORMATIVE QK 705-- The preceding error occurred at field {text}, key {text}, file {file}.

Condition Identifier: QKE\$LINDEXED_SEQUENTIAL Code: QK 705

Description: There was an error in your file at the location indicated in this message.

User Action: Refer to the preceding error message to determine the cause of the error. Use the Quick editor to correct the error.

--INFORMATIVE QK 800-- {file} is not a profile.

Condition Identifier: QKE\$PNOT_PROFILE Code: QK 800

Description: You are specifying a file that is not a profile. Execute the command again and enter the name of a file that is a profile. If the profile is in listable format, use option F on the UTILITY FUNCTIONS. For more information, refer to the Quick Online Manual using the index topic 'Profile' or 'Utility'.

User Action: Not applicable.

--INFORMATIVE QK 801-- {file} is not a profile; its file attributes do not match a profile.

Condition Identifier: QKE\$PNOT_PROFILE_CONTENTS Code: QK 801

Description: While executing the QUICK command, or while retrieving a profile in the UTILITY screen, you specified a file that is not a profile.

User Action: Execute the command again, and specify a file that is a profile. For more information refer to the Quick Online Manual using the index topic 'Profile' 'Utility'.

--INFORMATIVE QK 802-- {file} is incompatible with the current version of Quick.

Condition Identifier: QKE\$PINCOMPATIBLE_PROFILE Code: QK 802

Description: You tried to use a profile which was created by another version of Quick and cannot be read by the current version of Quick.

User Action: For a discussion of how to convert your profile from one version to another, refer to the Quick Online Manual using the index topics 'copying your profile to a listable file' and 'loading a new profile'.

--INFORMATIVE QK 803-- Your profile may not be a NOS/VE standard file.

Condition Identifier: QKE\$PROFILE_NULL_DEVICE Code: QK 803

Description: The file that you specified for your profile is \$NULL or a NOS/VE standard file such as \$INPUT.

User Action: Specify a file that is either an existing profile or a new file that is neither \$NULL nor a NOS/VE standard file.

--INFORMATIVE QK 804-- Your profile may not be assigned to your terminal.

Condition Identifier: QKE\$PROFILE_ASSIGN_TERMINAL Code: QK 804

Description: The file that you specified for your profile is assigned to your terminal.

User Action: Specify a file that is not assigned to your terminal.

--INFORMATIVE QK 807-- File is unknown. Description was not obtained.

Condition Identifier: QKE\$PAUX_NEW_FILE Code: QK 807

Description: While in the UTILITY FUNCTIONS window, you tried to retrieve a description from a file that does not exist or is not available to your job.

User Action: Execute the command again, and enter the name of a file that exists. For more information, see the Quick Online Manual using the index topic 'utility'.

--INFORMATIVE QK 808-- Description was not obtained.

Condition Identifier: QKE\$PDESCR_NOT_GOTTEN Code: QK 808

Description: While in the UTILITY FUNCTIONS window, you unsuccessfully tried to retrieve a description from a previous message which was diagnosed as unsuccessful.

User Action: Correct the previous error.

--ERROR QK 809-- Your error file may not be a NOS/VE standard file.

Condition Identifier: QKE\$ERROR_FILE_NULL_DEVICE Code: QK 809

Description: The file that you specified for your error file is \$NULL or a NOS/VE standard file such as \$INPUT.

User Action: Specify an error file that is neither \$NULL nor a NOS/VE standard file.

--INFORMATIVE QK 810-- File is not a profile. Description was not obtained.

Condition Identifier: QKE\$PAUX_NOT_PROFILE Code: QK 810

Description: While in the UTILITY FUNCTIONS window, you tried to retrieve a description from a file that is not a profile.

User Action: Execute the command agains, and enter the name of a file that is a profile. For more information, refer to the Quick Online Manual using the index topic 'utility'.

--ERROR QK 811-- Your error file may not be assigned to your terminal.

Condition Identifier: QKE\$ERROR_FILE_ASSIGN_TERMINAL Code: QK 811

Description: The file that you specified for your error file is assigned to your terminal.

User Action: Specify a file that is not assigned to your terminal.

--INFORMATIVE QK 812-- Profile is incompatible with this version. Description was not obtained.

Condition Identifier: QKE\$PAUX_INCOMPATIBLE_PROFILE Code: QK 812

Description: In the UTILITY screen, you tried to get a description from a profile which was created by another version of Quick. This profile cannot be read by the current version.

User Action: For a discussion of how to convert your profile from one version to another, refer to the Quick Online Manual using the index topics 'Copying your profile to a listable file' and 'Loading a new profile'.

--INFORMATIVE QK 814-- Output description was not obtained.

Condition Identifier: QKE\$POUT_DESCR_NOT_GOTTEN Code: QK 814

Description: In the UTILITY screen, you unsuccessfully attempted to get an output description from a file. A previous message diagnosed why you were unsuccessful.

User Action: Correct the previous error.

--INFORMATIVE QK 816-- File is unknown. Output description was not obtained.

Condition Identifier: QKE\$PAUX_NEW_FILE_OUT Code: QK 816

Description: While in the UTILITY FUNCTIONS window, you tried to retrieve an output description from a file that does not exist or is not available to your job.

User Action: Execute the command again, and enter the name of a file that exists. For more information, see the Quick Online Manual using the index topic 'utility'.

--INFORMATIVE QK 820-- Profile is incompatible with this version. Output description not obtained.

Condition Identifier: QKE\$PAUX_INCOMPAT_PROFILE_OUT Code: QK 820

Description: In the UTILITY screen you tried to get an output description from a profile that was created by another version of Quick. This output description cannot be read by the current version of Quick.

User Action: For a discussion of how to convert your profile from one version to another, refer to the Quick Online Manual using the index topics 'Copying your profile to a listable file' and 'Loading a new profile'.

--INFORMATIVE QK 822-- Error in compilation of profile. Output description was not obtained.

Condition Identifier: QKE\$P_ERR_PROFILE_NOT_OUTOD Code: QK 822

Description: While in the UTILITY FUNCTIONS window, you tried to retrieve an output description from another file. There was an error in the profile you tried to retrieve.

User Action: Correct the profile before trying to retrieve the output description. You can switch to the other profile within the UTILITY FUNCTIONS window by executing option A.

--INFORMATIVE QK 826-- Out-file description was not written.

Condition Identifier: QKE\$POD_NOT_PRINT Code: QK 826

Description: In the UTILITY screen, you unsuccessfully attempted to write an output description to a file. A previous message diagnosed why you were unsuccessful.

User Action: Correct the previous error.

--INFORMATIVE QK 832-- Profile was not written.

Condition Identifier: QKE\$PPROFILE_NOT_PRINT Code: QK 832

Description: In writing your profile to a 'write profile' format file, there was an error and the file was not written. An example would be an invalid file name. A previous message diagnosed the error.

User Action: Correct the previous error.

--INFORMATIVE QK 834-- Profile was not loaded.

Condition Identifier: QKESP_NOT_LOAD Code: QK 834

Description: In loading your profile from a 'write profile' format file, there was an error and the profile was not loaded. Examples would be a file not there or an invalid file name. A previous message diagnosed the error.

User Action: Correct the previous error.

--INFORMATIVE QK 836-- File was empty. Profile was not loaded.

Condition Identifier: QKESP_EMPTY_NOT_LOAD Code: QK 836

Description: While in the UTILITY FUNCTIONS window, you tried to load a profile from an empty file.

User Action: Execute the command again, and enter the name of a file that was written by the 'write profile' utility function. For more information, refer to the Quick Online Manual using the index topic 'utility'.

--INFORMATIVE QK 838-- First line was not properly formatted. Profile was not loaded.

Condition Identifier: QKESP_FORMAT_NOT_LOAD Code: QK 838

Description: While in the UTILITY FUNCTIONS window, you tried to load a profile from a file that was not in 'write profile' format.

User Action: Execute the command again, and enter the name of a file that is in 'write profile' format. For more information, refer to the Quick Online Manual, using the index topic 'utility'.

--INFORMATIVE QK 840-- Profile load terminated prematurely.

Condition Identifier: QKESP_LOAD_PREMATURE_END Code: QK 840

Description: An error diagnosed in the preceding error message stopped the profile load.

User Action: Correct the previous error.

--INFORMATIVE QK 842-- Errors and listable output were written to {file}.

Condition Identifier: QKESP_ERRORS_LISTABLE_OUTPUT Code: QK 842

Description: Quick detected one or more errors while loading a profile.

User Action: View the error file with an editor and take action according to the errors on the error file.

--INFORMATIVE QK 848-- Profile was not copied.

Condition Identifier: QKESP_PROFILE_NOT_COPIED Code: QK 848

Description: In copying your profile to another profile, there was an error and the profile was not copied. An example would be an invalid file name. A previous message diagnosed the error.

User Action: Correct the previous error.

--INFORMATIVE QK 850-- Profile was not changed.

Condition Identifier: QKESP_PROFILE_UNCHANGED Code: QK 850

Description: In the UTILITY screen, you tried to switch to another profile. A previous message diagnosed why you were unsuccessful.

User Action: Correct the previous error.

--INFORMATIVE QK 856-- The profile documentation was not changed.

Condition Identifier: QKESP_DOCUMENT_UNCHANGED Code: QK 856

Description: While in the UTILITY FUNCTIONS, you pressed 'D' to document the profile, then pressed ABANDON.

User Action: No action necessary.

--INFORMATIVE QK 866-- Switch to profile {file} completed.

Condition Identifier: QKESP_SWITCH_COMPLETE Code: QK 866

Description: In the UTILITY FUNCTIONS, you successfully executed option A, and switched to another profile.

User Action: No action necessary.

--INFORMATIVE QK 868-- Description obtained from profile {file}.

Condition Identifier: QKESP_DESCR_GOTTEN Code: QK 868

Description: In the UTILITY FUNCTIONS window, you successfully executed option B, get description.

User Action: No action necessary.

--INFORMATIVE QK 870-- Profile copied to file {file}.

Condition Identifier: QKESP_PROFILE_COPIED Code: QK 870

Description: In the UTILITY screen, you successfully executed option C, Copy current profile to another file.

User Action: No action necessary.

--INFORMATIVE QK 872-- Profile written on file {file}.

Condition Identifier: QKESP_PROFILE_PRINTED Code: QK 872

Description: Your profile was written to a file in 'write profile' format.

User Action: No action required.

--INFORMATIVE QK 874-- Output description written on file {file}.

Condition Identifier: QKESP_OD_PRINTED Code: QK 874

Description: In the UTILITY screen, you successfully executed option H, Write output description onto a file in COBOL format.

User Action: No action necessary.

--INFORMATIVE QK 876-- Profile loaded from file {file}.

Condition Identifier: QKESP_PROFILE_LOADED Code: QK 876

Description: In the UTILITY screen, you successfully executed option F, Load profile from file in 'write profile' format.

User Action: No action necessary.

--INFORMATIVE QK 878-- Output description obtained from profile {file}.

Condition Identifier: QKESP_OD_GOTTEN Code: QK 878

Description: In the UTILITY screen, you successfully executed option G, Get output description.

User Action: No action necessary.

--INFORMATIVE QK 882-- *ERROR* Skipping over excess lines.

Condition Identifier: QKESP_EXCESS_LINES Code: QK 882

Description: There is an error in your listable profile: there are extra lines. As an example, there are eight lines in the footing window in the listable profile although a footing window may only contain six lines.

User Action: You can correct the listable profile in an editor by deleting the extra lines, or you can use Quick to correct the profile after loading is complete. If you fix it in Quick, notice that Quick reads in the correct portions of the listable profile, and only omits the extra lines.

--INFORMATIVE QK 884-- *ERROR* Skipping over filter which is not in numerical order.

Condition Identifier: QKESP_INCORRECT_FILTER Code: QK 884

Description: There is an error in your listable profile: a filter is out of order.

User Action: You can correct the listable profile in an editor by putting the filter in order, or you can use Quick to correct the profile after loading is complete. If you fix it in Quick, notice that Quick reads in the correct portions of the listable profile and only omits the unknown option.

--INFORMATIVE QK 886-- *ERROR* Misplaced SELECT OPTION window ignored. Searching for next window.

Condition Identifier: QKES\$P_MISPLACED_SELECT_REJECT Code: QK 886

Description: There is an error in your listable profile: a SELECT/REJECT window is misplaced.

User Action: You can correct the listable profile with an editor, or you can use Quick to correct the profile after loading is complete. If you fix it in Quick, notice that Quick reads in the correct portions of the listable profile, and you need only examine the window that was missing in the listable profile.

--INFORMATIVE QK 888-- *ERROR* Searching for first window heading.

Condition Identifier: QKES\$P_SEARCH_FOR_1ST_WINDOW Code: QK 888

Description: There is an error in your listable profile: a heading is missing.

User Action: You can correct the listable profile in an editor, or you can use Quick to correct the profile after loading is complete. If you fix it in Quick, notice that Quick reads in the correct portions of the listable profile, and you need only examine the window that was missing in the listable profile.

--INFORMATIVE QK 892-- *ERROR* Unknown option ignored.

Condition Identifier: QKES\$P_UNKNOWN_OPTION_IGNORED Code: QK 892

Description: There is an error in your listable profile: there is an unknown option. As an example, there is an extra OUTPUT OPTION in the listable profile.

User Action: You can correct the listable profile in an editor by deleting the unknown option, or replacing it with a known option, or you can use Quick to correct the profile after loading is complete. If you fix it in Quick, notice that Quick reads in the correct portions of the listable profile and only omits the unknown option.

--INFORMATIVE QK 894-- You have insufficient access permission to save changes to your profile. You may make changes to the profile, but these changes will not be saved when you exit Quick or switch to another profile.

Condition Identifier: QKES\$P_INSUFFICIENT_PERM Code: QK 894

Description: You must have READ, APPEND, MODIFY, and SHORTEN permissions to save changes to your profile.

User Action: If you need to save changes to your profile, get these permissions.

--INFORMATIVE QK 900-- {text} requires {text} spaces. You need {text} more spaces after {text}.

Condition Identifier: QKES\$RUSER_SYMBOL_BLANKS Code: QK 900

Description: There is not enough space within 800 columns for the information you specified. As an example, you entered \$DATE in column 776. Since \$DATE requires 10 spaces, you will get this error message saying that you need five more spaces after \$DATE.

User Action: Allow more space. In the example, you would move \$DATE five spaces to the left.

--INFORMATIVE QK 902-- This report specification is not within a field.

Condition Identifier: QKES\$RIN_NO_FIELD Code: QK 902

Description: Self-explanatory.

User Action: Move the report specification so that it is within the appropriate field. For more information, refer to the Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 904-- This report specification is within more than one field.

Condition Identifier: QKES\$RIN_MULTIPLE_FIELDS Code: QK 904

Description: The specification in the REPORT BREAKS AND TOTALS field lies under more than one field.

User Action: Delete or insert the necessary characters to align the report specification within the field. For more information, refer to the Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 906-- This is the second break ordinal you have specified for this field.

Condition Identifier: QKE\$RMULTIPLE_ORDINAL Code: QK 906

Description: In the REPORT BREAKS window, you have placed more than one number (separated by a blank) under one field. An example would be: '1 2'.

User Action: Remove all but one number such as '1' or '2'. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 908-- This character is not a valid report specification character.

Condition Identifier: QKE\$RINVALID_CHAR Code: QK 908

Description: Self-explanatory.

User Action: Delete the invalid character or replace it with a valid character. For more information, refer to The Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 914-- You may not specify both a break ordinal and statistics for this field.

Condition Identifier: QKE\$RBREAK_WITH_TOTAL Code: QK 914

Description: Self-explanatory.

User Action: Remove either the break ordinal or the t. For more information refer to the Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 916-- You may not specify P without a break ordinal.

Condition Identifier: QKE\$RPRINT_WITHOUT_BREAK Code: QK 916

Description: In the REPORT BREAKS AND TOTALS window, you are entering a P without a number indicating the hierarchy of the break.

User Action: Insert a number or delete the P. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 918-- C for count is the only type of statistics that you may specify for a nonnumeric field.

Condition Identifier: QKE\$RTOTAL_NONNUMERIC Code: QK 918

Description: You asked for totals on a non-numeric field.

User Action: Remove the t from the report specification, or move it to a numeric field. For more information, refer to the Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 920-- This is the second time you have specified this type of statistics for this field.

Condition Identifier: QKE\$RMULTIPLE_TOTAL Code: QK 920

Description: In the REPORT BREAKS window, you have placed more than one T under a single field, for example: 'tt' or 'ttt'.

User Action: Remove all but one T: 't'. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 922-- This is the second time you have specified P for this field.

Condition Identifier: QKE\$RMULTIPLE_PRINT Code: QK 922

Description: In the REPORT BREAKS window, you have placed more than one P under a single field. Examples would be: '1pp' or 'pp'.

User Action: Remove all but one P: '1p'. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 926-- For a statistics only report to a file, you must have a break.

Condition Identifier: QKE\$RTOF_NO_BREAK Code: QK 926

Description: In the OUTPUT OPTIONS, you have selected 'write to file' or 'write to DIF file', and in the REPORT menu, you have selected TOTALS ONLY=Y with no control breaks.

User Action: Include a control break, change to TOTALS ONLY=n, or change the OUTPUT OPTION to 'Display at terminal' 'Write to listable file'. For more information, refer to the Quick Online Manual using the index topic 'report' or 'output file'.

--INFORMATIVE QK 928-- You have specified a statistics only report and no statistics.

Condition Identifier: QKE\$RTOTALS_ONLY_NO_TOTALS Code: QK 928

Description: In the REPORT window, you have specified 'Y' on TOTALS ONLY, and you have not included any Ts in the REPORT BREAKS AND TOTALS.

User Action: Enter at least one T or change TOTALS ONLY to N. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 930-- You may not specify statistics for an excluded field.

Condition Identifier: QKE\$REXCLUDE_TOTAL Code: QK 930

Description: In the REPORT BREAKS AND TOTALS window, you are entering T on a field that is not included in the ORDER OF FIELDS.

User Action: Include the field in the ORDER OF FIELDS or remove the T from the REPORT BREAKS AND TOTALS window. For more information, refer to the Quick Online Manual using the index topic 'Fields' or 'Totals'.

--INFORMATIVE QK 934-- This field must contain a positive integer.

Condition Identifier: QKE\$RPAGE_EJECT_ON_INTEGER Code: QK 934

Description: Self-explanatory.

User Action: Replace the invalid characters with a positive integer. For more information, refer to the Quick Online Manual using the index topic 'Page eject'.

--INFORMATIVE QK 936-- You did not specify this control break ordinal in the REPORT BREAKS AND STATISTICS window.

Condition Identifier: QKE\$RUNKNOWN_BREAK Code: QK 936

Description: You entered a number in the PAGE EJECT option that you did not include in the REPORT BREAKS window.

User Action: Enter the number in the REPORT BREAKS window or delete it from the PAGE EJECT option. For more information, refer to the Quick Online Manual using the index topic 'Report'.

--INFORMATIVE QK 938-- You may not page eject on the most minor break in a statistics only report.

Condition Identifier: QKE\$REJECT_BREAK_TOTALS_ONLY Code: QK 938

Description: Self-explanatory.

User Action: Change the TOTALS ONLY option or the number in the PAGE EJECT ON CONTROL BREAK option. For more information, refer to the Quick Online Manual using the index topic 'Totals'.

--INFORMATIVE QK 940-- You must specify an IN file and DESCRIPTION before report fields.

Condition Identifier: QKE\$R_NO_IN_FILE Code: QK 940

Description: You are trying to specify breaks and totals before specifying the INPUT file and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data' or 'Utility'.

--INFORMATIVE QK 942-- You must specify a DESCRIPTION before report fields.

Condition Identifier: QKE\$R_NO_DATA_DESCR Code: QK 942

Description: You are trying to create a REPORT before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from another profile through the UTILITY function. For more information, refer to the Quick Online Manual using the index topic 'Description of data' or 'Utility'.

--INFORMATIVE QK 1002-- This select specification is not within a field.

Condition Identifier: QKE\$SEIN_NO_FIELD Code: QK 1002

Description: Self-explanatory.

User Action: Move the specification so that it is within the appropriate field. For more information, refer to the Quick Online Manual using the index topic 'Selection screen'.

--INFORMATIVE QK 1004-- This select specification is within more than one field.

Condition Identifier: QKE\$SEIN_MULTIPLE_FIELDS Code: QK 1004

Description: The specification in the SELECTION VALUES window lies under more than one field.

User Action: Delete or insert the necessary characters to align the specification within one field. For more information, refer to the Quick Online Manual using the index topic 'Selection screen'.

--INFORMATIVE QK 1006-- This token is invalid.

Condition Identifier: QKE\$SEINVALID_TOKEN Code: QK 1006

Description: You have entered an invalid character or combination of characters in the SELECTION VALUES window. Examples would be '=c' under a numeric field or '=12.b'.

User Action: Replace the invalid characters with valid characters, for example '=3' or '=123'. For more information, refer to the Quick Online Manual using the index topic 'Select'.

--INFORMATIVE QK 1008-- This expression ended prematurely.

Condition Identifier: QKE\$SEPREMATURE_END Code: QK 1008

Description: An expression in your SELECTION VALUES window contains only an operator. Examples would be '=' or '>.'.

User Action: Finish the expression by placing a value after the operator. For more information refer to the Quick Online Manual using the index topic 'Select'.

--INFORMATIVE QK 1010-- The select/reject window must contain S or R.

Condition Identifier: QKE\$SEINVALID_SELECT_REJECT Code: QK 1010

Description: Self-explanatory.

User Action: Replace the invalid character with S or R. For more information, refer to the Quick Online Manual using the index topic 'reject'.

--INFORMATIVE QK 1012-- This is the second selection value in one line for one field.

Condition Identifier: QKE\$SEMULTIPLE_VALUES Code: QK 1012

Description: Self-explanatory.

User Action: Delete one selection value or move it to another line or field. For more information, refer to the Quick Online Manual using the index topic 'Formulas'.

--INFORMATIVE QK 1014-- You must specify a DESCRIPTION before you may select.

Condition Identifier: QKE\$SENO_DATA_DESCR Code: QK 1014

Description: You are trying to SELECT records before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from aFor more information, refer to the Quick Online Manual using the index topic 'Description of data' or 'Utility'. UTILITY function. r more information

--INFORMATIVE QK 1016-- When you use wild card symbols, you must use the = or <> relation.

Condition Identifier: QKESSEWILD_CARD_NOT_EQUAL Code: QK 1016

Description: You have used a wild card in the SELECTION VALUES WINDOW WITH A RELATION OTHER THAN = OR <>. For example: '<?at' or '>rat*'.
User Action: Use = or <> instead: '<>?at' or '=rat*'. For more information, refer to the Quick Online Manual using the index topic 'Wild card'.

--INFORMATIVE QK 1018-- Internal error -- missing range keys.

Condition Identifier: QKESSEMISSING_RANGE_KEYS Code: QK 1018

Description: You have encountered an error within Quick.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 1020-- Internal error -- invalid range in range-table.

Condition Identifier: QKESSE RANGE_ERROR Code: QK 1020

Description: You have encountered an error within Quick.

User Action: Follow site-defined procedures for reporting the software error.

--INFORMATIVE QK 1022-- The SELECT_FILTER subcommand is not available with this menu.

Condition Identifier: QKESSE_SELECT_NOT_AVAILABLE Code: QK 1022

Description: Self-explanatory.

User Action: To get a list of Quick functions and subcommands available with the current menu, press HELP, then enter MENU at the home line. The SELECT_FILTER command is available with the Main menu, the Edit menu, and the Edit Select menu.

--INFORMATIVE QK 1024-- The largest filter number you may specify now is {text}.

Condition Identifier: QKESSE_LARGEST_FILTER Code: QK 1024

Description: You selected a filter number that was too high. The largest filter you may select is one higher than the number of filters you have defined.

User Action: Select a filter no higher than the number indicated in the error message. For more information, refer to the Quick Online Manual using the index topic 'Filter'.

--INFORMATIVE QK 1026-- You must specify an IN file and DESCRIPTION before you may select.

Condition Identifier: QKESSE_NO_IN_FILE Code: QK 1026

Description: You are trying to specify selection criteria before specifying the INPUT field and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data' or 'Utility'.

--INFORMATIVE QK 1105-- This sort specification is not within a field.

Condition Identifier: QKES\$IN_NO_FIELD Code: QK 1105

Description: The specification in the SORT KEYS window lies between fields.

User Action: Move the specification so that it lies under one field. For more information, refer to the Quick Online Manual using the index topic 'Sort key'.

--INFORMATIVE QK 1110-- This sort specification is within more than one field.

Condition Identifier: QKES\$IN_MULTIPLE_FIELDS Code: QK 1110

Description: The specification in the SORT KEYS window lies under more than one field.

User Action: Delete or insert the necessary characters to align the sort specification within one field. For more information, refer to the Quick Online Manual using the index topic 'Sort key'.

--INFORMATIVE QK 1115-- You have specified more than one sort key order for this field.

Condition Identifier: QKE\$SMULTIPLE_ORDER Code: QK 1115

Description: Self-explanatory.

User Action: Delete the extra characters so that there is only one sort key order per field. For more information, refer to the Quick Online Manual using the index topic 'Sort key'.

--INFORMATIVE QK 1120-- You have specified more than one sort key ordinal for this field.

Condition Identifier: QKE\$SMULTIPLE_ORDINAL Code: QK 1120

Description: Self-explanatory.

User Action: Delete the extra sort key ordinals so that there is only one sort key ordinal per field. For more information, refer to the Quick Online Manual using the index topic 'Sort key'.

--INFORMATIVE QK 1125-- The sort key ordinal (an integer) optionally followed by the sort key order (A or D) is expected here.

Condition Identifier: QKE\$SINVALID_CHAR Code: QK 1125

Description: You have entered an invalid SORT key. An example would be 1A.

User Action: Correct the key as described in the error message. For more information, refer to the Quick Online Manual using the index topic 'Sort key'.

--INFORMATIVE QK 1130-- The sort key ordinal (an integer) must precede the sort key order.

Condition Identifier: QKE\$SORDINAL_MISSING Code: QK 1130

Description: You have specified a sort key order, A or D, but no sort key ordinal.

User Action: Specify the sort key ordinal followed by the sort key order, for example '1A'.

--INFORMATIVE QK 1135-- You must specify an IN file and DESCRIPTION before you may sort.

Condition Identifier: QKE\$S_NO_IN_FILE Code: QK 1135

Description: You are trying to specify sort keys before specifying the INPUT file and describing data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data' or 'Utility'.

--INFORMATIVE QK 1140-- You must specify a DESCRIPTION before you may sort.

Condition Identifier: QKE\$S_NO_DATA_DESCR Code: QK 1140

Description: You are trying to SORT records before describing the data.

User Action: Specify a data description by using DESCR or by bringing it from another profile through the UTILITY function. For more information, refer to the Quick Online Manual using the index topic 'Description of data' or 'Utility'.

--INFORMATIVE QK 1200-- Specify data description, then view.

Condition Identifier: QKE\$VNEED_DESCRIPTION Code: QK 1200

Description: You are pressing VIEW before you have entered the description of data.

User Action: Press DESCR and enter the description. For more information, refer to the Quick Online Manual using the index topic 'description of data.'

--INFORMATIVE QK 1202-- You must specify an IN file and a DESCRIPTION before you may view.

Condition Identifier: QKE\$VPRESS_IN_DESCR Code: QK 1202

Description: You are trying to VIEW a file before specifying the INPUT file and describing the data.

User Action: Specify the IN file, then the DESCRIPTION OF DATA. For more information, refer to the Quick Online Manual using the index topic 'Input file' or 'Description of data' or 'Utility'.

--INFORMATIVE QK 1204-- You cannot view a file which is assigned to your terminal.

Condition Identifier: QKE\$V_CANT_VIEW_FILE Code: QK 1204

Description: You assigned your input file to your terminal. You cannot view this file.

User Action: Specify an input file which is not assigned to your terminal.

--INFORMATIVE QK 1205-- You cannot view a NOS/VE standard file.

Condition Identifier: QKE\$V_CANT_VIEW_NULL_FILE Code: QK 1205

Description: You specified \$NULL or a NOS/VE standard file such as \$INPUT as your INPUT FILE NAME.

User Action: Specify an input file which is neither \$NULL nor a NOS/VE standard file.

--INFORMATIVE QK 1208-- File {file} empty

Condition Identifier: QKE\$VFILE_EMPTY Code: QK 1208

Description: You are trying to VIEW a file that is empty.

User Action: No action necessary.

--ERROR QK 1300-- Quick requires a terminal with at least 80 columns.

Condition Identifier: QKE\$SC80_COLUMNS_NEEDED Code: QK 1300

Description: Your terminal definition says your terminal has fewer than 80 columns.

User Action: Use another terminal or another terminal definition.

--INFORMATIVE QK 1301-- Quick terminated.

Condition Identifier: QKE\$COMMAND_TERMINATED Code: QK 1301

Description: Quick terminated because it received a terminal interrupt such as terminate_break, pause_break, or terminal_connection_broken.

User Action: Restart Quick if you wish.

--ERROR QK 1302-- Quick requires a terminal with at least 24 lines.

Condition Identifier: QKE\$SC24_LINES_NEEDED Code: QK 1302

Description: Your terminal definition says your terminal has fewer than 24 lines.

User Action: Use another terminal or another terminal definition.

--ERROR QK 1303-- The {text} application menu on your help module contains {text} menu items. It may only contain {text} menu items.

Condition Identifier: QKE\$SC_MENU_MISMATCH Code: QK 1303

Description: The application menu that appears on your help module contains menu items which do not belong there.

User Action: Correct the application menu by using the correct menu items.

--ERROR QK 1304-- Your terminal is not set up for full screen usage. You must have specified your terminal name using SETTA TRM=name, and you must have a terminal definition for the terminal you are using.

Condition Identifier: QKE\$SCNO_LINE_TERMINAL Code: QK 1304

Description: Your terminal is not set up for full screen usage.

User Action: Get a full screen terminal definition and, before running Quick, specify your terminal model by entering: "setta trm='model'"; as an example: "setta trm=cdc721" or "setta trm=z29".

--INFORMATIVE QK 1310-- Put cursor in a window whose contents may be shifted right and left (for example, TITLE window) before performing this function.

Condition Identifier: QKE\$SC_CURSOR_IN_LONG_WINDOW Code: QK 1310

Description: You have attempted the C> or C< function when the cursor was not in a window whose contents may be shifted right and left.

User Action: Put the cursor in a window whose contents may be shifted right and left, then perform the C> or C< function.

--INFORMATIVE QK 1312-- Put cursor in the View window before performing this function.

Condition Identifier: QKE\$SC_CURSOR_IN_VIEW_WINDOW Code: QK 1312

Description: You have attempted the UP or DOWN function when the cursor was not in the View window whose contents may be shifted up or down.

User Action: Put the cursor in the View window, then perform the UP and DOWN function.

--INFORMATIVE QK 1314-- Put cursor in the Execute results window before performing this function.

Condition Identifier: QKE\$SC_CURSOR_IN_EXECUTE_WINDOW Code: QK 1314

Description: You have attempted the UP or DOWN function but the cursor was not in the Execute results windows whose contents may be shifted up and down.

User Action: Put the cursor in the Execute results window, then perform the UP or DOWN function.

--INFORMATIVE QK 1316-- Put cursor in the {text} window before performing this function.

Condition Identifier: QKE\$SC_CURSOR_IN_EDIT_WINDOW Code: QK 1316

Description: You have attempted the UP or DOWN function but the cursor was not in the EDIT FILE window whose contents may be shifted up and down.

User Action: Put the cursor in the EDIT FILE window, then perform the UP or DOWN function.

--INFORMATIVE QK 1318-- You may only modify the record marked for modification.

Condition Identifier: QKE\$SC_TYPE_IN_PROTECTED_AREA Code: QK 1318

Description: You are trying to modify a record before marking it for modification.

User Action: Position the cursor anywhere on the record you want to modify and press the UPDATE function key, or press the INSERT function key to insert a new record. For more information, refer to the Quick Online Manual using the index topics 'Update' and 'Insert'.

--INFORMATIVE QK 1320-- This window may have a maximum of {text} lines.

Condition Identifier: QKE\$SC_WINDOW_MAX_LINES Code: QK 1320

Description: You are trying to insert too many lines. For example, you are trying to insert more than six lines in the FOOTING window.

User Action: No action necessary. You must fit the information in the number of lines available. For more information, refer to the Quick Online Manual using the index topic 'Insert line'.

--INFORMATIVE QK 1322-- This window may have a minimum of {text} lines.

Condition Identifier: QKE\$SC_WINDOW_MIN_LINES Code: QK 1322

Description: You are trying to delete too many lines from a window. For example, in the FOOTING window, you are trying to delete all of the lines, and there must be at least one line in this window.

User Action: No action necessary. You must leave the lines there.

--INFORMATIVE QK 1324-- Quick subcommands and SCL commands are not allowed at this time.

Condition Identifier: QKES\$SC_LOCKOUT_SUB_CMDS Code: QK 1324

Description: You entered a command at the home line when it was not allowed. As an example, this would be when Quick is waiting for you to press OK or CANCEL.

User Action: Perform the action that Quick is waiting for, then enter the home line command.

--INFORMATIVE QK 1332-- You may only insert a line in the {text} window.

Condition Identifier: QKES\$SC_INSERT_LINE_WINDOW Code: QK 1332

Description: Self-explanatory.

User Action: If you are trying to insert a line, move the cursor to the appropriate window, then press INS L.

--INFORMATIVE QK 1334-- You may only insert a line in the TITLE or COLUMN HEADINGS window.

Condition Identifier: QKES\$SC_INSERT_LINE_WINDOWS Code: QK 1334

Description: Self-explanatory.

User Action: If you are trying to insert a line in the TITLE or COLUMN HEADINGS window, move the cursor to that window, then press INS L.

--INFORMATIVE QK 1336-- You may not insert a line within any window on this screen.

Condition Identifier: QKES\$SC_INSERT_LINE_NO_WINDOW Code: QK 1336

Description: Self-explanatory.

User Action: Do not try to insert lines on this screen. For more information, refer to the Quick Online Manual using the index topic 'Insert lines'.

--INFORMATIVE QK 1338-- You may only delete a line from the {text} window.

Condition Identifier: QKES\$SC_DELETE_LINE_WINDOW Code: QK 1338

Description: Self-explanatory.

User Action: If you are trying to delete a line, move the cursor to the appropriate window, then press DEL L.

--INFORMATIVE QK 1340-- You may only delete a line from the TITLE or COLUMN HEADINGS window.

Condition Identifier: QKES\$SC_DELETE_LINE_WINDOWS Code: QK 1340

Description: Self-explanatory.

User Action: If you are trying to delete a line from the TITLE or COLUMN HEADINGS window, move the cursor to that window, then press DEL L.

--INFORMATIVE QK 1342-- You may not delete a line from any window on this screen.

Condition Identifier: QKES\$SC_DELETE_LINE_NO_WINDOW Code: QK 1342

Description: Self-explanatory.

User Action: Do not try to delete lines in this screen. For more information, refer to the Quick Online Manual using the index topic 'Delete line'.

--INFORMATIVE QK 1344-- Unknown input received from terminal.

Condition Identifier: QKES\$SC_UNKNOWN_INPUT_RECEIVED Code: QK 1344

Description: There was garbage on the communication line.

User Action: You may have to re-enter the last command if it did not execute properly.

--INFORMATIVE QK 1346-- Input buffer filled by too many typed characters before {text}.

Condition Identifier: QKE\$SC_INPUT_BUFFER_OVERFLOW Code: QK 1346

Description: You typed too many characters before pressing RETURN and lost some characters that did not fit in an internal buffer.

User Action: Use the Clear function to refresh your screen so you know what characters were lost. Enter the characters again.

--INFORMATIVE QK 1348-- BKW is only active in the SELECT, EDIT, VIEW, or XECUTE screen.

Condition Identifier: QKE\$SC_BKW_NOT_ACTIVE Code: QK 1348

Description: Self-explanatory.

User Action: To use BKW, first press SELECT, EDIT, VIEW, or XECUTE.

--INFORMATIVE QK 1350-- LINDN (Line-Down) only moves a line down in the EDIT, VIEW, or XECUTE screen.

Condition Identifier: QKE\$SC_DOWN_NOT_ACTIVE Code: QK 1350

Description: Self-explanatory.

User Action: To use DOWN, first press EDIT, VIEW, or EXECUTE.

--INFORMATIVE QK 1352-- FWD is only active in the SELECT, EDIT, VIEW, or XECUTE screen.

Condition Identifier: QKE\$SC_FWD_NOT_ACTIVE Code: QK 1352

Description: Self-explanatory.

User Action: To use FWD, first press SELECT, EDIT, VIEW, or XECUTE.

--INFORMATIVE QK 1354-- LINUP (Line-Up) only moves a line up in the EDIT, VIEW, or XECUTE screen.

Condition Identifier: QKE\$SC_UP_NOT_ACTIVE Code: QK 1354

Description: Self-explanatory.

User Action: To use UP, first press EDIT, VIEW, or XECUTE.

--INFORMATIVE QK 1356-- Use the INSLIN function to insert a line at the bottom of the window.

Condition Identifier: QKE\$SC_USE_FUNCTION Code: QK 1356

Description: You attempted to insert a line at the bottom of the window by pressing your terminal's INSERT LINE key.

User Action: Use Quick's INS L function.

--INFORMATIVE QK 1358-- {text} is an unknown command. After returning to Quick, you may enter MENU at the home line for a list of available Quick functions and subcommands.

Condition Identifier: QKE\$SC_UNKNOWN_COMMAND Code: QK 1358

Description: You entered an unknown command at the home line.

User Action: Enter a correct command.

--INFORMATIVE QK 1360-- You entered a home line command and pressed a function key. The home line command will be processed and the function key ignored.

Condition Identifier: QKE\$SC_IGNORE_FUNCTION Code: QK 1360

Description: Self-explanatory.

User Action: If you need to execute the function key that you pressed, press it again after the home line command is processed.

--INFORMATIVE QK 1402-- Your profile specifies that output will be displayed at the terminal; therefore, you must run Quick in interactive mode.

Condition Identifier: QK\$EXONLY_SHOW Code: QK 1402

Description: You are running Quick in execute only mode, but your profile specifies that output will be displayed at your terminal.

User Action: Either use a different profile, or else run Quick in interactive mode.

--ERROR QK 1404-- Your profile has not been initialized; therefore, you must run Quick in interactive mode.

Condition Identifier: QK\$EXONLY_NEW_PROFILE Code: QK 1404

Description: You are running Quick in execute only mode using a profile which has never been initialized.

User Action: Either use a different profile, or else run Quick in interactive mode.

--ERROR QK 1406-- An error has occurred while executing Quick in execute_only mode. You must rerun in interactive mode, specifying the same files that you used in execute_only mode, for Quick to diagnose and for you to correct the error.

Condition Identifier: QK\$EXONLY_ANY_ERROR Code: QK 1406

Description: You were executing Quick in execute-only mode. There is an error in your profile. For example, there is an error in the file description or there is an invalid calculation formula.

User Action: Rerun QUICK in interactive mode. If you specified FROM or TO files, change the INPUT and OUTPUT file names in the profile to match these files. Press XECUTE and the error will be diagnosed in interactive mode. Correct the errors.

--ERROR QK 1407-- The UID and UPW parameters are required when accessing IM/DM in execute_only mode.

Condition Identifier: QK\$SEO_UID_UPW_PARAMETERS Code: QK 1407

Description: You attempted to run Quick in execute_only mode using a profile that accesses IM/DM. However, you did not specify your IM/DM user identification code and user password with the UID and UPW parameters, which are required when accessing IM/DM.

User Action: Start Quick again, and include the UID and UPW parameters as follows:

```
UID=your_id UPW=your_pw
```

--INFORMATIVE QK 1562-- MENU

Condition Identifier: QK\$W_MENU Code: QK 1562

Description: In your sequential file, you are trying to move or calculate a value which does not fit in the field you described. For example, you could be trying to store 1234 in a field described as 999.

User Action: Examine the error message to determine which field is too short. Either increase the field length or modify the calculation that caused this error. For more information, refer to the Quick Online Manual using the index topic 'Description of data' and 'Calculation window'.

--INFORMATIVE QK 1924-- {text}

Condition Identifier: QK\$DM_LONG_TEXT Code: QK 1924

Description: Quick received an error message from IM/DM. The message you received is the exact message that Quick received from IM/DM. No further information is available.

User Action: No further information is available.

--INFORMATIVE QK 1934-- Your profile specifies database access, but IM/Quick is unable to access the IM/DM interface. If IM/DM is available at your installation, enter the command `:family_name.DMROOT.DM$COMMANDS` to make the IM/DM interface available. Family_name is the name of the family in which IM/DM is installed, usually NVE. Then restart IM/Quick.

Condition Identifier: QKE\$DM_CANNOT_LOAD Code: QK 1934

Description: You attempted to run Quick using a profile that accesses IM/DM. However, the IM/DM interface is currently not in place or IM/DM is not available at your installation.

User Action: If your site has IM/DM, type the following command before restarting Quick. Replace FAMILY_NAME with the family in which IM/DM is installed:

```
:FAMILY_NAME.dmroot.dm$commands
```

For example, IM/DM is usually installed in the NVE family. If this is true for your site, type the following command, then start Quick:

```
:nve.dmroot.dm$commands
```

If your site does not have IM/DM, you cannot use a profile that accesses IM/DM. Restart Quick using a different profile.

--ERROR QK 1935-- Your profile specifies database access, but IM/Quick is unable to access the IM/DM interface. If IM/DM is available at your installation, enter the command `:family_name.DMROOT.DM$COMMANDS` to make the IM/DM interface available. Family_name is the name of the family in which IM/DM is installed, usually NVE. Then restart IM/Quick.

Condition Identifier: QKE\$DM_CANNOT_LOAD_E Code: QK 1935

Description: You attempted to run Quick using a profile that accesses IM/DM. However, the IM/DM interface is currently not in place or IM/DM is not available at your installation.

User Action: If your site has IM/DM, type the following command before restarting Quick. Replace FAMILY_NAME with the family in which IM/DM is installed:

```
:FAMILY_NAME.dmroot.dm$commands
```

For example, IM/DM is usually installed in the NVE family. If this is true for your site, type the following command, then start Quick:

```
:nve.dmroot.dm$commands
```

If your site does not have IM/DM, you cannot use a profile that accesses IM/DM. Restart Quick using a different profile.

--INFORMATIVE QK 2077-- The LINUP (Line-Up) function is only active for the EDIT, VIEW, CALC, and XECUTE functions. It moves the line at which the cursor is positioned to the top of the display area on the screen for these functions.

Condition Identifier: QKE\$H_UP_NOOP_FUNCTION Code: QK 2077

Description: Self-explanatory.

User Action: Replace the invalid character with Y or N. For more information, refer to the Quick Online Manual using the index topic 'Totals only'.

--INFORMATIVE QK 2604-- {text} is neither a Quick function within the current menu nor a Quick subcommand; therefore, there is no help available for {text}.

To get a list of the Quick functions and subcommands for which help is available, enter MENU at the home line and press {text}.

Condition Identifier: QKE\$H_UNKNOWN_SUB_CMD Code: QK 2604

Description: While in help mode, you entered an unknown command at the home line.

User Action: Enter a known command.

--FATAL RF 1-- An invalid data fragment has been detected by {text}.

Condition Identifier: RFE\$INVALID_DATA_FRAGMENT Code: RF 1

Description: A network application has called RHFAM/VE with an invalid data fragment. This may be an invalid pointer or a pointer to a segment with the wrong read/write privileges to be used as a data buffer.

User Action: Notify your site analyst.

--ERROR RF 2-- File {file} existed prior to the attempt to establish a connection.

Condition Identifier: RFE\$FILE_ALREADY_EXISTS Code: RF 2

Description: RHFAM/VE connections are associated with a local file. This file may not exist when requesting a connection.

User Action: Notify your site analyst.

--ERROR RF 3-- Connection {file} is not established. Server has not responded.

Condition Identifier: RFE\$CONNECT_IN_PROGRESS Code: RF 3

Description: RHFAM/VE has been called with a connection request that is invalid because the remote server application has not responded to the establish connection request.

User Action: Notify your site analyst.

--FATAL RF 4-- This job is not registered as the RHFAM initiated server, {text}.

Condition Identifier: RFE\$NOT_REGISTERED_SERVER Code: RF 4

Description: The server instance that is attempting to sign on to use RHFAM/VE services is not the server instance that was initiated by RHFAM/VE.

User Action: Notify your site analyst.

--ERROR RF 5-- Unable to switch connection file {file} because the file is open.

Condition Identifier: RFE\$FILE_NOT_CLOSED Code: RF 5

Description: A connection file must be closed before it can be switched to another job.

User Action: Notify your site analyst.

--FATAL RF 6-- A network block has been received that exceeds the requested size, requested size : {text}, received size : {text}.

Condition Identifier: RFE\$NETWORK_BLOCK_EXCEEDED Code: RF 6

Description: A network block has been received from the LCN that exceeds the data block size that was requested. This may be caused by an interface error between the two communicating network applications.

User Action: Notify your site analyst.

--FATAL RF 7-- A record mode network block has been received that is smaller than the requested size and does not have a termination mark.

Condition Identifier: RFE\$PARTIAL_NETWORK_BLOCK Code: RF 7

Description: A network block has been received from the LCN that is not a full network block. LCN protocol requires that record mode data blocks must be full or have a termination mark associated with the network block.

User Action: Notify your site analyst.

--INFORMATIVE RF 8-- No connection available for application {text}.

Condition Identifier: RFE\$CONNECTION_NOT_AVAILABLE Code: RF 8

Description: The application request to acquire a connection from RHFAM/VE failed because there is no connection available.

User Action: Notify your site analyst.

--FATAL RF 9-- The maximum active connection limit has been reached for application {text}.

Condition Identifier: RFE\$MAX_APPL_CONNECTS_EXCEEDED Code: RF 9

Description: The current limit on active connections allowed for all instances of an application has been reached. This is a temporary condition.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 10-- An invalid connection event has been specified for await connection event.

Condition Identifier: RFE\$INVALID_CONNECTION_EVENT Code: RF 10

Description: An invalid parameter has been specified for the request to await for an RHFAM/VE connection event.

User Action: Notify your site analyst.

--ERROR RF 11-- Connection {file} was terminated.

Condition Identifier: RFE\$CONNECTION_TERMINATED Code: RF 11

Description: The current application request to RHFAM/VE was terminated abnormally because the connection was terminated in the LCN network.

User Action: Notify your site analyst,

--FATAL RF 12-- Connection limit does not match defined value for {text}.

Condition Identifier: RFE\$MAX_CONNECTION_MISMATCH Code: RF 12

Description: A server application was started by RHFAM/VE that was supposed to reserve a certain number of connections. When this server signed on to RHFAM/VE it did not reserve the correct number of connections.

User Action: Notify your site analyst.

--ERROR RF 13-- Host {text} is not defined in the RHFAM configuration.

Condition Identifier: RFE\$DESTINATION_HOST_UNDEFINED Code: RF 13

Description: An application has attempted to request a connection to a remote host that is not defined in the active LCN network configuration.

User Action: Notify your site analyst.

--ERROR RF 14-- Application {text} is defined but not active.

Condition Identifier: RFE\$APPL_NOT_ACTIVE Code: RF 14

Description: An application that has been defined via MANRN but has not been activated via MANRN has attempted to use RHFAM/VE services.

User Action: Notify your site analyst.

--FATAL RF 15-- Specifying {text} connections on the sign-on request exceeds the defined limit for {text}.

Condition Identifier: RFE\$DEFINED_CONNECTS_EXCEEDED Code: RF 15

Description: An application has attempted to sign on to use RHFAM/VE services and has requested to use more connections than are available for this server. This should be a temporary condition.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 16-- An invalid application kind has been specified for {text}.

Condition Identifier: RFE\$INVALID_APPLICATION_KIND Code: RF 16

Description: An application has attempted to sign on to RHFAM/VE as an invalid application kind.

User Action: Notify your site analyst.

--ERROR RF 21-- All paths to {text} are currently down.

Condition Identifier: RFE\$PATHS_TO_DESTINATION_DOWN Code: RF 21

Description: There is currently a problem with the LCN hardware connections to the desired destination. A problem may exist in the LCN or the destination host may not be available. This is a temporary problem.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 26-- Connection {file} was rejected because host {text} does not currently have the resources to satisfy the request.

Condition Identifier: RFE\$REMOTE_HOST_BUSY Code: RF 26

Description: The requested destination host does not currently have the needed resources to satisfy a connection request. This is a temporary condition.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 27-- Server {text} is not defined locally to RHFAM.

Condition Identifier: RFE\$REMOTE_SERVER_UNDEFINED Code: RF 27

Description: An application has requested an LCN connection to a server on the local host. However, the requested server is not defined on the local host.

User Action: Notify your site analyst.

--ERROR RF 32-- {text} request terminated because the transfer timeout was exceeded.

Condition Identifier: RFE\$TRANSFER_TIMEOUT Code: RF 32

Description: The connection has been terminated because no activity has been detected on the connection file for a period of time greater than the requested timeout value. This could indicate a problem with the remote host.

User Action: Notify your site analyst.

--ERROR RF 34-- {text} request failed because the job is not signed on as application {text}.

Condition Identifier: RFE\$NOT_SIGNED_ON Code: RF 34

Description: A request has been made to RHFAM/VE by an application that has not signed on as the specified application. An application must sign on to RHFAM/VE before requesting services.

User Action: Notify your site analyst.

--ERROR RF 35-- Connection {file} was rejected by host {text} with an invalid reject code: {text}.

Condition Identifier: RFE\$UNKNOWN_REJECT_CODE Code: RF 35

Description: A connection request to a remote host was rejected with a reject code that is not a currently supported LCN reject code.

User Action: Notify your site analyst.

--FATAL RF 37-- {text} failed because application {text} has reached its connection limit.

Condition Identifier: RFE\$EXCEEDED_CONNECT_LIMIT Code: RF 37

Description: An application has attempted to obtain more connections than it reserved when it signed on the RHFAM/VE.

User Action: Notify your site analyst.

--ERROR RF 38-- There is no path defined to remote server {text} on remote host {text}.

Condition Identifier: RF\$PATH_TO_REMOTE_UNDEFINED Code: RF 38

Description: No possible connection path has been defined through the LCN network that would allow a connection to the desired application on the desired remote host. This could be a problem with the local LCN configuration definition.

User Action: Notify your site analyst.

--FATAL RF 39-- {text} failed because the job is not signed on as server application {text}.

Condition Identifier: RF\$NOT_SIGNED_ON_AS_SERVER Code: RF 39

Description: An application has attempted to request a service from RHFAM/VE that is restricted to a server application and the application is not signed on as a server application kind.

User Action: Notify your site analyst.

--FATAL RF 40-- {text} failed because this job is not signed on as an RHFAM application.

Condition Identifier: RF\$NOT_AN_RHFAM_JOB Code: RF 40

Description: An application has requested a service from RHFAM/VE without first signing on.

User Action: Notify your site analyst.

--ERROR RF 41-- An invalid attribute value has been specified for {text}.

Condition Identifier: RF\$INVALID_ATTRIBUTE_VALUE Code: RF 41

Description: An application has requested to set or change a connection file attribute to an invalid value.

User Action: Notify your site analyst.

--ERROR RF 42-- This job is already signed on as {text}.

Condition Identifier: RF\$ALREADY_SIGNED_ON Code: RF 42

Description: An application has attempted to sign on to RHFAM/VE that is already signed on.

User Action: Notify your site analyst.

--ERROR RF 43-- Application {text} could not sign-off because all connections have not been terminated.

Condition Identifier: RF\$CONNECTIONS_NOT_TERMINATED Code: RF 43

Description: An application has attempted to sign off from RHFAM/VE before terminating all connections.

User Action: Notify your site analyst.

--ERROR RF 44-- No server response has been received for connection {file}.

Condition Identifier: RF\$NO_SERVER_RESPONSE Code: RF 44

Description: There has been no response received from the remote application on the requested connection.

User Action: Notify your site analyst.

--ERROR RF 45-- Connection {file} was rejected by host {text} because server {text} had reached its connection limit.

Condition Identifier: RF\$SERVER_BUSY Code: RF 45

Description: The remote host rejected the connect request because the maximum server applications as defined on the remote host are currently in use. This is a temporary condition.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 46-- Connection {file} was rejected by host {text} because the remote RHFAM was terminating.

Condition Identifier: RFE\$REMOTE_RHF_SHUTDOWN Code: RF 46

Description: The remote host rejected the connect request because the remote RHFAM system is currently terminating. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 47-- Connection {file} was rejected by host {text} because the destination LID {text} is currently disabled.

Condition Identifier: RFE\$SERVER_LID_DISABLED Code: RF 47

Description: The remote host rejected the connect request because it currently has the requested location disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 48-- Connection {file} was rejected by server application {text} on host {text}.

Condition Identifier: RFE\$SERVER_REJECT_RESPONSE Code: RF 48

Description: The remote server application rejected the connect request.

User Action: Notify your site analyst.

--ERROR RF 49-- Connection {file} was rejected by host {text} because the password was not valid.

Condition Identifier: RFE\$PASSWORD_UNDEFINED Code: RF 49

Description: The remote host rejected the connect request because the LCN password sent with the connect request does not match the defined password on the remote host.

User Action: Notify your site analyst.

--ERROR RF 50-- Connection {file} was rejected by host {text} because client {text} is not defined.

Condition Identifier: RFE\$CLIENT_UNDEFINED Code: RF 50

Description: The remote host rejected the connect request because it does not have the application requesting the connection defined.

User Action: Notify your site analyst.

--ERROR RF 51-- Connection {file} was rejected by host {text} because server {text} is currently disabled.

Condition Identifier: RFE\$SERVER_DISABLED Code: RF 51

Description: The remote host rejected the connect request because it currently has the requested server disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 52-- Connection {file} was rejected by host {text} because client {text} is currently disabled.

Condition Identifier: RFE\$CLIENT_DISABLED Code: RF 52

Description: The remote host rejected the connect request because it currently has the application requesting the connection disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 53-- Connection {file} was rejected by host {text} because the source PID {text} is currently disabled.

Condition Identifier: RFE\$CLIENT_PID_DISABLED Code: RF 53

Description: The remote host rejected the connect request because it currently has the LCN physical identifier of the requesting host disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 54-- Connection {file} was rejected by host {text} because the source NAD {text} is currently disabled.

Condition Identifier: RFE\$CLIENT_NAD_DISABLED Code: RF 54

Description: The remote host rejected the connect request because it currently has the NAD of the requesting host disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 55-- Connection {file} was rejected by host {text} because the specified TCUs are currently disabled.

Condition Identifier: RFE\$TCU_DISABLED Code: RF 55

Description: The remote host rejected the connect request because it currently has the TCUs required to communicate with the requesting host disabled. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 56-- Connection {file} to host {text} failed because the remote RHFAM is not active.

Condition Identifier: RFE\$RHF_NOT_ACTIVE Code: RF 56

Description: The remote host rejected the connect request because the RHFAM system on the remote host is not active. This is a condition that may exist for an indefinite period of time.

User Action: Notify your site analyst if the condition persists.

--ERROR RF 57-- Connection {file} was rejected by host {text} because server {text} is not defined.

Condition Identifier: RFE\$SERVER_UNDEFINED Code: RF 57

Description: The remote host rejected the connect request because it currently does not have the requested server application defined.

User Action: Notify your site analyst.

--ERROR RF 58-- Connection {file} was rejected by host {text} because the server LID {text} is not defined.

Condition Identifier: RFE\$SERVER_LID_UNDEFINED Code: RF 58

Description: The remote host rejected the connect request because it currently does not have the logical identifier of the requested remote host defined.

User Action: Notify your site analyst.

--ERROR RF 59-- Connection {file} was rejected by host {text} because the source PID {text} is not defined.

Condition Identifier: RFE\$CLIENT_PID_UNDEFINED Code: RF 59

Description: The remote host rejected the connect request because it currently has the LCN physical identifier of the requesting host undefined.

User Action: Notify your site analyst.

--ERROR RF 60-- Connection {file} was rejected by host {text} because the source NAD {text} is not defined.

Condition Identifier: RFESCLIENT_NAD_UNDEFINED Code: RF 60

Description: The remote host rejected the connect request because it currently has the NAD of the requesting host undefined.

User Action: Notify your site analyst.

--ERROR RF 61-- Connection {file} was rejected by host {text} because the NAD access codes do not match.

Condition Identifier: RFESACCESS_CODE_INVALID Code: RF 61

Description: The remote host rejected the connect request because the NAD access code does not match the remote host NAD access code.

User Action: Notify your site analyst.

--ERROR RF 62-- Connection {file} was rejected by host {text} because the destination device is invalid.

Condition Identifier: RFESDEVICE_INVALID Code: RF 62

Description: The remote host rejected the connect request because the host connection code on the remote system does not match the requested host connection code.

User Action: Notify your site analyst.

--ERROR RF 63-- Connection {file} was rejected by host {text} because the specified TCUs were undefined or invalid.

Condition Identifier: RFESTCU_INVALID Code: RF 63

Description: The remote host rejected the connect request because it currently has the TCUs required to communicate with the requesting host undefined.

User Action: Notify your site analyst.

--WARNING RF 64-- The requested event is not available.

Condition Identifier: RFESNO_AVAILABLE_EVENT Code: RF 64

Description: An application was waiting for an expected event and it did not occur.

User Action: Notify your site analyst.

--ERROR RF 65-- Connection {file} does not exist.

Condition Identifier: RFESCONNECTION_NOT_ACTIVE Code: RF 65

Description: An application attempted an RHFAM/VE connection file request on a connection file that did not exist.

User Action: Notify your site analyst.

--ERROR RF 69-- {text} failed because RHFAM is not active.

Condition Identifier: RFESSYSTEM_TASK_NOT_ACTIVE Code: RF 69

Description: The application request to RHFAM/VE failed because the RHFAM/VE system task is not active.

User Action: Notify your site analyst.

--ERROR RF 70-- {text} request failed because connection file {file} is not a RHFAM device class file.

Condition Identifier: RFESFILE_DEVICE_CLASS_NOT_RHF Code: RF 70

Description: An application attempted to use a file identifier in an RHFAM/VE request that identifies a file that is not a connection file.

User Action: Notify your site analyst.

--ERROR RF 74-- A network block sequence error has been detected on connection {file}. Expected block = {text}, Received block = {text}.

Condition Identifier: RF\$BLOCK_SEQUENCE_ERROR Code: RF 74

Description: A network data block was received from the LCN that is out of sequence.

User Action: Notify your site analyst.

--ERROR RF 75-- Request failed because connection {file} is waiting for an accept or reject.

Condition Identifier: RF\$CONNECTION_WAITING_ACCEPT Code: RF 75

Description: An application attempted to use a request other than accept connection or reject connection on a connection file that is waiting for an application accept or reject.

User Action: Notify your site analyst.

--ERROR RF 76-- Connection {file} has been rejected.

Condition Identifier: RF\$CONNECTION_REJECTED Code: RF 76

Description: An application attempted to use a connection file in an RHFAM/VE request that has been rejected. The only valid request on a rejected connection file is a terminate connection.

User Action: Notify your site analyst.

--ERROR RF 77-- A {text} block was received during a {text} transfer on connection {file}.

Condition Identifier: RF\$RECEIVE_MODE_CONFLICT Code: RF 77

Description: A receive data request from the LCN failed because a record mode data block was received when a message mode network block was requested or a message mode data block was received when a record mode data block was requested.

User Action: Notify your site analyst.

--WARNING RF 78-- Connection {file} has been offered for switching to job {text}.

Condition Identifier: RF\$SWITCH_OFFERED Code: RF 78

Description: An application has attempted an RHFAM/VE request on a connection file that has been offered for switching to another local job. The only valid request on a connection file offered for switching is a terminate connection or a terminate switch offer.

User Action: Notify your site analyst.

--INFORMATIVE RF 79-- Connection {file} has been switched and accepted by job {text}.

Condition Identifier: RF\$SWITCH_ACCEPTED Code: RF 79

Description: An application has attempted an RHFAM/VE request on a connection file that has been switched and accepted by another local job. The only valid request on a switched and accepted connection file is a terminate connection.

User Action: Notify your site analyst.

--INFORMATIVE RF 80-- Connection {file} is normal.

Condition Identifier: RF\$CONNECTED Code: RF 80

Description: An application has attempted an RHFAM/VE request on a connection file that is not valid when the connection is established.

User Action: Notify your site analyst.

--WARNING RF 81-- The switch offer on connection {file} has not been accepted.

Condition Identifier: RF\$SWITCH_OFFER_NOT_ACCEPTED Code: RF 81

Description: An application has asked RHFAM/VE to offer a connection file for switching to another local job and wait a specified time. The time has expired and the other local job did not accept the connection that was offered.

User Action: Notify your site analyst.

--WARNING RF 82-- There is no connection switch offer for application {text}.

Condition Identifier: RFE\$NO_SWITCH_OFFERED Code: RF 82

Description: An application has asked RHFAM/VE to accept a switch offer and wait a specified time if a switch offer is not available. The time has expired and no connection has been offered for switching.

User Action: Notify your site analyst.

--WARNING RF 83-- There is no switch offer on connection {file} to cancel.

Condition Identifier: RFE\$NO_SWITCH_OFFER_PENDING Code: RF 83

Description: An application has requested that RHFAM/VE cancel a connection file switch offer, but the specified connection file has not been offered for switching to another job.

User Action: Notify your site analyst.

--ERROR RF 84-- An invalid attribute key has been specified for {text}.

Condition Identifier: RFE\$INVALID_ATTRIBUTE_KEY Code: RF 84

Description: An application was requested to set, change, or get a connection file attribute and has specified an invalid attribute key.

User Action: Notify your site analyst.

--ERROR RF 85-- Connection {file} has been terminated by a shutdown of the RHFAM/VE system task.

Condition Identifier: RFE\$SYSTEM_TASK_SHUTDOWN Code: RF 85

Description: The specified connection file on an RHFAM/VE request has been terminated because of an RHFAM/VE system shutdown.

User Action: Notify your site analyst.

--ERROR RF 86-- Connection {file} has been terminated because of a local network access device failure.

Condition Identifier: RFE\$LOCAL_NAD_FAILURE Code: RF 86

Description: The specified connection file on an RHFAM/VE request has been terminated because of a local NAD failure.

User Action: Notify your site analyst.

--ERROR RF 87-- Connection {file} has been terminated because of a local system interrupt.

Condition Identifier: RFE\$SYSTEM_INTERRUPT Code: RF 87

Description: The specified connection file on an RHFAM/VE request has been terminated because of a local system interrupt.

User Action: Notify your site analyst.

--ERROR RF 88-- The attempted connection failed because local NAD {text} is currently saturated.

Condition Identifier: RFE\$LOCAL_NAD_BUSY Code: RF 88

Description: An attempted connection request failed because the local NAD has reached its connection limit.

User Action: Notify your site analyst.

--ERROR RF 89-- Request {text} failed because the local NAD is down.

Condition Identifier: RFE\$LOCAL_NAD_DOWN Code: RF 89

Description: An attempted connection request failed because the local NAD has been set to a DOWN or OFF condition.

User Action: Notify your site analyst.

--WARNING RF 91-- Host {text} is currently disabled in the RHFAM configuration.

Condition Identifier: RFE\$DESTINATION_HOST_DISABLED Code: RF 91

Description: An attempted connection request failed because the destination host has been set to an OFF state in the local LCN configuration.

User Action: Notify your site analyst.

--ERROR RF 92-- {text} failed because the job is not signed on as client application {text}.

Condition Identifier: RFE\$NOT_SIGNED_ON_AS_CLIENT Code: RF 92

Description: An application has attempted to request a service from RHFAM/VE that is restricted to a client application and the application is not signed on as a client application kind.

User Action: Notify your site analyst.

--WARNING RF 93-- All of the specified record mode data was not sent because {text}.

Condition Identifier: RFE\$UNABLE_TO_SEND_ALL_DATA Code: RF 93

Description: An application has attempted to send record mode data across the LCN that is not a multiple of network blocks and has no ending delimiter.

User Action: Notify your site analyst.

--ERROR RF 100-- The configuration file directives in {file} are not installable. Check the {file} file for errors.

Condition Identifier: RFE\$CONFIGURATION_FILE_ERROR Code: RF 100

Description: Errors have been detected in the input (LCN configuration directive file) to the MANRN subcommand INSTALL_RHFAM_CONFIGURATION. The new LCN configuration is not installable.

User Action: Refer to the output of the VERIFY_RHFAM_CONFIGURATION to locate the directives in error and correct the LCN configuration directive file.

--ERROR RF 101-- A local host directive has already been processed.

Condition Identifier: RFE\$DUPLICATE_LOCAL_HOST Code: RF 101

Description: An LCN configuration directive file may contain one and only one DEFINE_LOCAL_HOST directive.

User Action: Remove all the DEFINE_LOCAL_HOST directives except one.

--ERROR RF 102-- The element {text} was specified on another {text} definition.

Condition Identifier: RFE\$DUPLICATE_COMPONENT_NAME Code: RF 102

Description: Two or more elements in the LCN have been defined with the same name. All elements of the LCN must be defined with unique names.

User Action: Rename the elements with unique names.

--ERROR RF 104-- Host connection {text} on nad {text} is not valid for this request.

Condition Identifier: RFE\$INVALID_HOST_CONNECTION Code: RF 104

Description: An invalid host connection has been specified on a configuration directive.

User Action: Ensure that the specified host connection has been defined.

--ERROR RF 105-- Component {text} must be defined prior to use in a DEFINE_LCN_PATH directive.

Condition Identifier: RFE\$UNDEFINED_PATH_ELEMENT Code: RF 105

Description: LCN elements must be defined before they are used on the DEFINE_LCN_PATH directive.

User Action: Define all the LCN elements before using them on the DEFINE_LCN_PATH directive.

--ERROR RF 106-- Invalid path. NAD {text} and NAD {text} do not have a common trunk.

Condition Identifier: RF\$NO_TRUNK_MATCH Code: RF 106

Description: There are no common trunks defined between the two specified NADs. Therefore it is not possible to have a path between the two NADs.

User Action: Correct the DEFINE_LCN_PATH directive to specify the correct elements or insure that the elements have been correctly defined using either the DEFINE_LOCAL_NAD or DEFINE_REMOTE_NAD directives.

--FATAL RF 110-- The AUTO_PATH_GENERATION directive can only be specified once in a directives file.

Condition Identifier: RF\$DUPLICATE_AUTPG Code: RF 110

Description: Only one AUTO_PATH_GENERATION directive may be specified in an LCN configuration directive file.

User Action: Remove all but one AUTO_PATH_GENERATION directive from the configuration directive file.

--FATAL RF 111-- The AUTO_PATH_GENERATION directive can not be specified after a DEFINE_LCN_PATH directive.

Condition Identifier: RF\$AUTPG_AFTER_DEFLP Code: RF 111

Description: The use of the AUTO_PATH_GENERATION directive must precede the use of any DEFINE_LCN_PATH directives in the LCN configuration directive file.

User Action: Either move the AUTO_PATH_GENERATION directive before the DEFINE_LCN_PATH directive or remove the DEFINE_LCN_PATH directive.

--ERROR RF 112-- A minimum of one {text} definition is required.

Condition Identifier: RF\$REQUIRED_DEF_MISSING Code: RF 112

Description: A required LCN configuration directive was not found in the input configuration directives.

User Action: Add the missing directive to the LCN configuration directive file.

--ERROR RF 115-- Logical identifier {text} is defined more than once for the same host.

Condition Identifier: RF\$DUPLICATE_LID Code: RF 115

Description: The same logical identifier has been specified more than once for the same host.

User Action: Remove the redundant logical identifiers.

--CATASTROPHIC RF 132-- The peripheral request could not be completed because {text}; in routine {text}.

Condition Identifier: RF\$REQUEST_PROCESSING_ERR Code: RF 132

Description: An unrecoverable error has been detected by the LCN PP driver when communicating with the NAD.

User Action: Notify your site analyst.

--FATAL RF 150-- The RHFAM/VE system task must be initiated from the system job.

Condition Identifier: RF\$INVALID_TASK_ORIGIN Code: RF 150

Description: The RHFAM/VE system task may only be run as a task under the system job.

User Action: None.

--FATAL RF 151-- Only one copy of the RHFAM/VE system task can be running at any instant.

Condition Identifier: RF\$SYSTEM_TASK_RUNNING Code: RF 151

Description: Only one copy of the RHFAM/VE system task may be executing at any one point in time.

User Action: None.

--FATAL RF 152-- Unusable RHFAM/VE configuration file : {text}.

Condition Identifier: RFE\$INVALID_CONFIG_FILE Code: RF 152

Description: No installed configuration file has been found or a configuration file has been found that is incompatible with the current version of RHFAM/VE.

User Action: Reinstall the LCN configuration using the MANAGE_RHFAM_NETWORK subcommand INSTALL_RHFAM_CONFIGURATION.

--CATASTROPHIC RF 154-- The RHFAM system task failed to start a pp program.

Condition Identifier: RFE\$UNABLE_TO_START_A_PP Code: RF 154

Description: RHFAM/VE was not able to start the LCN PP driver.

User Action: Notify your site analyst.

--INFORMATIVE RF 159-- Microcode load {text} for {text} {text}.

Condition Identifier: RFE\$MICROCODE_LOADED Code: RF 159

Description: The NAD microcode was successfully loaded in the specified NAD.

User Action: None.

--INFORMATIVE RF 164-- {text} control message was received: nad={text}, my path={text}, his path={text}, block type={text}.

Condition Identifier: RFE\$UNEXPECTED_CONTROL_MESSAGE Code: RF 164

Description: An unexpected message has been received on an active connection by the RHFAM/VE system task.

User Action: Notify your site analyst.

--WARNING RF 165-- NAD {text} has trunk {text} on TCU {text} configured, but the TCU is not available.

Condition Identifier: RFE\$NAD_TCU_UNAVAILABLE Code: RF 165

Description: The specified NAD has the specified TCU disabled on the NAD or the specified TCU is not installed on the NAD.

User Action: Contact Engineering Services to verify the status of the NAD hardware and change the LCN configuration directives accordingly.

--WARNING RF 166-- NAD {text} is configured with address {text}, but {text} is the real address.

Condition Identifier: RFE\$NAD_ADDRESS_MISMATCH Code: RF 166

Description: The specified NAD is configured and defined with a NAD address that does not match the NAD hardware setting for the NAD address.

User Action: Contact Engineering Services to verify the hardware setting for the NAD address and correct the LCN configuration to define the correct NAD address.

--FATAL RF 171-- NAD {text} failed device test.

Condition Identifier: RFE\$TEST_NAD_FAILURE Code: RF 171

Description: The specified NAD failed during a test of its memory and/or functions.

User Action: Contact Engineering Services to verify the status of the NAD hardware.

--ERROR RF 200-- File {file} must have page width of at least {text} characters.

Condition Identifier: RFE\$ILLEGAL_OUTPUT_FILE Code: RF 200

Description: The output file specified for the MANAGE_RHFAM_NETWORK displays must have a page width of at least 72 characters.

User Action: Change the file attributes of the output file or choose a different output file.

--ERROR RF 202-- Unable to find element {text}.

Condition Identifier: RFE\$ELEMENT_NOT_FOUND Code: RF 202

Description: The specified element on a MANRN display subcommand was not found.

User Action: Correct the element name on the MANRN display subcommand or display all elements to determine the correct element name.

--ERROR RF 203-- Unable to find logical identifier {text}.

Condition Identifier: RFE\$LOGICAL_ID_NOT_FOUND Code: RF 203

Description: The specified logical identifier was not found.

User Action: Correct the specified logical identifier or use the full display to determine the current logical identifiers in the configuration.

--ERROR RF 204-- Unable to find remote host {text}.

Condition Identifier: RFE\$PHYSICAL_ID_NOT_FOUND Code: RF 204

Description: The specific physical identifier was not found.

User Action: Correct the specified physical identifier or use the full display to determine the current physical identifiers in the configuration.

--ERROR RF 205-- The NAD dump file, {file} is empty.

Condition Identifier: RFE\$NAD_DUMP_FILE_EMPTY Code: RF 205

Description: The specified dump file contains no data.

User Action: Check the name of the dump file for errors or specify another dump file.

--ERROR RF 207-- File {file} is not an RHFAM configuration file.

Condition Identifier: RFE\$NOT_AN_RHFAM_CONFIG_FILE Code: RF 207

Description: The currently installed configuration file is not a valid RHFAM/VE configuration file.

User Action: Reinstall the RHFAM/VE configuration.

--ERROR RF 209-- Unable to change state of DOWN NAD {text}.

Condition Identifier: RFE\$DOWN_NAD_NOT_CHANGABLE Code: RF 209

Description: A DOWN state indicates that RHFAM/VE is performing maintenance activity on the specified NAD. This state cannot be changed until RHFAM/VE has completed its activity.

User Action: Wait for RHFAM/VE to complete its maintenance activity. If successful, the NAD will be returned to the ON state. If the state goes to OFF, contact Engineering Services to repair the NAD.

--ERROR RF 211-- Illegal to DOWN remote NADs.

Condition Identifier: RFE\$CANNOT_DOWN_REMOTE_NAD Code: RF 211

Description: Only local NADs may be set to the DOWN state.

User Action: None.

--ERROR RF 212-- Key value {text} must be alone in list when specifying {text}.

Condition Identifier: RFE\$TOO_MANY_KEY_VALUES Code: RF 212

Description: A list of keys has been specified on an MANRN subcommand that requires a key value or list of string values.

User Action: Specify only one key value or specify a list of string values.

--INFORMATIVE RF 213-- {text} duplicate lines are suppressed.

Condition Identifier: RFE\$SKIPPED_LINES Code: RF 213

Description: Duplicate lines have been suppressed in the output file at this point.

User Action: None.

--ERROR RF 214-- Unable to find application definition for {text}.

Condition Identifier: RFE\$APPL_NOT_DEFINED Code: RF 214

Description: The specified application is not defined.

User Action: Check the APPLICATION_NAME parameter.

--WARNING RF 215-- The application {text}, is already active.

Condition Identifier: RFE\$APPL_ALREADY_ACTIVE Code: RF 215

Description: The specified application is already active.

User Action: Check the APPLICATION_NAME parameter.

--WARNING RF 216-- The application {text}, is already inactive.

Condition Identifier: RFE\$APPL_ALREADY_INACTIVE Code: RF 216

Description: The specified application is already inactive.

User Action: Check the APPLICATION_NAME parameter.

--ERROR RF 217-- The client application {text}, is already defined as a server.

Condition Identifier: RFE\$CLIENT_DEFINED_AS_SERVER Code: RF 217

Description: The specified client is already defined as a server. Clients and servers must have unique names.

User Action: Rename either the client or the server.

--ERROR RF 218-- The application {text}, is already defined.

Condition Identifier: RFE\$DUPLICATE_APPL_DEFINITION Code: RF 218

Description: The specific application is already defined.

User Action: Either delete the present application definition or check the correctness of the APPLICATION_NAME parameter.

--ERROR RF 219-- Cannot delete active application definition {text}.

Condition Identifier: RFE\$APPL_NOT_INACTIVE Code: RF 219

Description: An active application may not be deleted.

User Action: Deactivate the application using DEACTIVATE_RHFAM_APPLICATION before deleting the application.

--ERROR RF 220-- Cannot delete application definition {text}, while job signed on.

Condition Identifier: RFE\$APPL_JOB_SIGNED_ON Code: RF 220

Description: It is not possible to delete an application while the application is actively using RHFAM/VE.

User Action: Terminate the active application before deleting the application.

--ERROR RF 221-- The server application {text}, is already defined as a client.

Condition Identifier: RFE\$SERVER_DEFINED_AS_CLIENT Code: RF 221

Description: The specified server is already defined as a client. Clients and servers must have unique names.

User Action: Rename either the server or the client.

--ERROR RF 222-- Job file not specified for rhfam_initiated server {text}.

Condition Identifier: RFE\$SERVER_JOB_NOT_SPECIFIED Code: RF 222

Description: No server job file has been specified.

User Action: Specify a server job file or change the server attribute of the RHFAM initiated server to FALSE.

--FATAL RF 223-- Caller not validated to {text}.

Condition Identifier: RFE\$CALLER_NOT_PRIVILEGED Code: RF 223

Description: The caller does not have the required privileges to perform the requested operation.

User Action: Contact the family administrator to obtain the required privileges.

--FATAL RF 224-- NAD {text} can only be tested when OFF.

Condition Identifier: RFE\$IMPROPER_NAD_STATE_FOR_TEST Code: RF 224

Description: The NAD must be turned off before it can be tested.

User Action: Using the MANAGE_REMOTE_NETWORK command CHANGE_NAD_STATE, change the state of the desired NAD to off and then reinitate the test.

--FATAL RF 225-- NAD {text} is currently being tested.

Condition Identifier: RFE\$TEST_ALREADY_IN_PROCESS Code: RF 225

Description: Only one NAD test can be in progress at a time.

User Action: Wait for the NAD test already in progress to complete before starting another NAD test.

--ERROR RH 0-- The NOS/170 file specified could not be found.

Condition Identifier: RHE\$MISSING_FILE Code: RH 0

Description: The file that you attempted to get (with the GET_FILE command) could not be found on the 170 side.

User Action: Log in to the 170 side to see if the file exists. If it does, contact a site analyst.

--ERROR RH 1-- The NOS/170 file specified is busy.

Condition Identifier: RHE\$CYCLE_BUSY Code: RH 1

Description: The file requested is attached to another job.

User Action: Try again at another time.

--ERROR RH 2-- A NOS/170 i/o error occurred processing this file.

Condition Identifier: RHE\$FILE_IO_ERROR Code: RH 2

Description: An error occurred transferring this file to or from mass storage.

User Action: Retry command. If the problem continues, notify the site analyst.

--ERROR RH 3-- The NOS/170 limit of permanent files has been exceeded.

Condition Identifier: RHE\$TOO_MANY_PERMANENT_FILES Code: RH 3

Description: You have exceeded the number of permanent files allowed.

User Action: Purge some permanent files or see the site analyst to raise the number of permanent files allowed.

--ERROR RH 4-- The NOS/170 limit for permanent file storage has been exceeded.

Condition Identifier: RHE\$TOO_MUCH_PF_STORAGE Code: RH 4

Description: The user's catalog limit for permanent file storage was exceeded.

User Action: Purge some permanent files, or see the site analyst to raise the permanent file storage limit.

--ERROR RH 5-- The length of the file exceeds the NOS/170 limit allowed.

Condition Identifier: RHE\$PERMANENT_FILE_TOO_LARGE Code: RH 5

Description: The file being transferred to NOS/170 is longer than the allowed NOS limit.

User Action: Break the file into smaller files or see the site analyst to raise the NOS limit.

--ERROR RH 6-- An error internal to remote host has occurred.

Condition Identifier: RHE\$RH_SYSTEM_ERROR Code: RH 6

Description: The remote host had a problem processing this command.

User Action: Retry command. If the problem continues, notify the site analyst.

--ERROR RH 7-- The amount of mass storage required is unavailable.

Condition Identifier: RHE\$MASS_STORAGE_UNAVAILABLE Code: RH 7

Description: The NOS/170 SYSTEM_TRACK limit has been reached.

User Action: Notify the site analyst.

--ERROR RH 8-- An error occurred referencing the NOS/170 file.

Condition Identifier: RHE\$FILE_ERROR Code: RH 8

Description: The NOS/170 file could not be referenced. The user may not have access to the file.

User Action: See the owner of the file to get access to the file and repeat the command.

--ERROR RH 9-- Unable to communicate with NOS/170 Remote Host.

Condition Identifier: RHE\$UNABLE_TO_COMMUNICATE Code: RH 9

Description: Communication between the 180 task and the 170 partner job has been lost.

User Action: Inform your site analyst and ask him/her to check the 170 system dayfile and the 180 NOS/VE job log.

--ERROR RH 10-- The user terminated this command with a break.

Condition Identifier: RHE\$TERMINAL_BREAK_OCCURRED Code: RH 10

Description: The user terminated the previous command.

User Action: None.

--ERROR RH 11-- Not able to communicate with the NOS/170 partner job.

Condition Identifier: RHE\$PARTNER_JOB_NOT_EXECUTING Code: RH 11

Description: The NOS/170 partner job is not executing. It may have aborted because of invalid information in the user's link attributes.

User Action: Execute the CHANGE_LINK_ATTRIBUTES command with valid information and try again. Request the system operator to restart IRHF170. If this still does not work, contact your site analyst.

--ERROR RH 12-- Replace_file at other than \$boi not allowed, file {text}.

Condition Identifier: RHE\$OPEN_POSITION_CONFLICT Code: RH 12

Description: To replace a file the NOS/VE file must be positioned at \$boi.

User Action: Rewind the file and repeat the command.

--ERROR RH 13-- IRHF170 was unable to route to nos's output queue the nos/ve print file {text}.

Condition Identifier: RHE\$UNABLE_TO_ROUTE_FILE Code: RH 13

Description: The NOS/VE print file could not be routed to NOS's output queue

User Action: Notify the site analyst.

--ERROR RH 14-- Irhf encountered error trying to transfer a print file for user {text}. The file was purged.

Condition Identifier: RHE\$UNABLE_TO_COMPLETE_TRANSFER Code: RH 14

Description: The print file to be transferred could not be attached by IRHF without errors and was purged from NOS/VE's output queue.

User Action: Notify the site analyst if the problem continues.

--ERROR RH 15-- The file {text} was purged from NOS's queue.

Condition Identifier: RHE\$PURGED_FILE Code: RH 15

Description: IRHF was unable to route the NOS/VE job to the NOS/VE input queue because of an invalid LOGIN card.

User Action: Correct the LOGIN card and resubmit the job.

--ERROR RH 16-- The transfer wanted cannot be done due to no memory link table space being available.

Condition Identifier: RHE\$NO_ML_FREE_ENTRIES_FOUND Code: RH 16

Description: A memory link table has become full and will not accept the request at this time.

User Action: Retry command. If the problem persists, inform your site analyst.

--ERROR RH 17-- The transfer wanted cannot be done due to an unexpected internal system status {text}.

Condition Identifier: RHE\$UNEXPECTED_ML_ERROR Code: RH 17

Description: An unexpected memory link error has occurred.

User Action: Inform your site analyst.

--ERROR RH 18-- The system is not running NOS or NOS/BE dual state, so the GET_FILE and REPLACE_FILE commands are not available.

Condition Identifier: RHE\$NO_PARTNER_EXISTS Code: RH 18

Description: The system is running NOS/VE, but dual state commands such as GET_FILE and REPLACE_FILE are not available on non-dual state systems.

User Action: None.

--ERROR RH 19-- A problem occurred during file transfer. Contact your site analyst.

Condition Identifier: RHE\$MISMATCHING_CODE Code: RH 19

Description: A problem occurred during a file transfer through the memory link.

User Action: Save the job log and the file being transferred and contact your site analyst.

--ERROR RH 20-- The {text} partner job is already signed on to the memory link but by a different application name. Retry the command or contact your site analyst.

Condition Identifier: RHE\$RECEIVER_ALREADY_SIGNED_ON Code: RH 20

Description: The dual state partner job is trying to sign on to the memory link, but is already signed on by a different application name.

User Action: Try the command again, or contact your site analyst.

--ERROR RH 21-- A problem occurred on the {text} side - check your {text} validations or your {text} file being busy.

Condition Identifier: RHE\$RECEIVER_NOT_SIGNED_ON Code: RH 21

Description: The dual state partner job could not get signed on to the memory link due either to incorrect validations or the 170 file being busy.

User Action: Verify your validations on the 170 side. Verify that the file is not busy. If the problem persists, contact your site analyst.

--ERROR RH 22-- A pause break was received during GET_FILE or REPLACE_FILE execution. The file transfer was terminated.

Condition Identifier: RHE\$PAUSE_BREAK_RECEIVED Code: RH 22

Description: A pause break was entered during execution of the GET_FILE or REPLACE_FILE commands. This causes termination of the file transfer command.

User Action: None.

--ERROR RH 23-- A terminal disconnect was received during GET_FILE or REPLACE_FILE execution. The file transfer was terminated.

Condition Identifier: RHE\$TERMINAL_CONNECTION_BROKEN Code: RH 23

Description: A terminal disconnect was received during execution of a GET_FILE or REPLACE_FILE command. The file transfer was terminated.

User Action: None.

--ERROR RH 24-- No LINK_USER entry has been defined for family ({text}).

Condition Identifier: RHE\$LUD_CANNOT_BE_FOUND Code: RH 24

Description: A LINK_USER entry with the specified family is not defined in the LINK_USER_DESCRIPTOR table.

User Action: Execute the CHANGE_LINK_ATTRIBUTES command with the correct family and try again.

--ERROR RH 50-- The NOS/BE permanent file OWNER IDENTIFICATION (ID) is in error.

Condition Identifier: RHE\$ID_ERROR Code: RH 50

Description: You may not replace (REPL) a new file with ID=SYSTEM or ID=PUBLIC.

User Action: Do not use ID=SYSTEM or ID=PUBLIC from NOS/VE.

--ERROR RH 51-- The NOS/BE local file name (LFN) is unknown.

Condition Identifier: RHE\$UNKNOWN_LFN Code: RH 51

Description: The file that you are trying to print via the PRINT_FILE command or trying to receive via the GET_FILE command is unknown.

User Action: Verify that the file exists and try again. If the problem persists, notify your site analyst.

--ERROR RH 52-- The NOS/BE permanent file catalog (PFC) is full.

Condition Identifier: RHE\$PFC_FULL Code: RH 52

Description: The permanent file catalog table is full.

User Action: Notify the system analyst. Remove some permanent files from the NOS/BE default permanent file set using PURGE or DUMPF.

--ERROR RH 53-- The NOS/BE file is not cataloged.

Condition Identifier: RHE\$FILE_NOT_CATALOGED Code: RH 53

Description: The permanent file does not exist in NOS/BE default permanent file set.

User Action: None.

--ERROR RH 54-- The NOS/BE file cycle limit (999) was reached.

Condition Identifier: RHE\$CYCLE_NUMBER_OVER_999 Code: RH 54

Description: The maximum cycle number was reached on REPF.

User Action: In NOS/BE, PURGE existing cycles of the permanent file or use RENAME to change the cycle numbers.

--ERROR RH 55-- The NOS/BE permanent file directory (PFD) is full.

Condition Identifier: RHE\$PFD_FULL Code: RH 55

Description: The permanent file directory table is full.

User Action: Notify the system analyst. Remove some permanent files from the NOS/BE default permanent file set using PURGE or DUMPF.

--ERROR RH 57-- The NOS/BE file cycle is incomplete or was dumped.

Condition Identifier: RHE\$CYCLE_INCOMPLETE Code: RH 57

Description: Access to this permanent file is not allowed because no PFC entry exists for this file.

User Action: Purge the permanent file and recreate using CATALOG or LOADPF.

--ERROR RH 58-- The NOS/BE permanent file being referenced is archived.

Condition Identifier: RHE\$FILE_ARCHIVED Code: RH 58

Description: The file has been saved on tape.

User Action: Attach the file on the NOS/BE system to force a reload from the tape.

--ERROR RH 59-- An illegal character in one of the parameters was specified.

Condition Identifier: RHE\$ILL_CHAR_IN_FDB_PARAM Code: RH 59

Description: A non-alphanumeric character was specified in a parameter passed to the NOS/BE permanent file processor.

User Action: Change the permanent file parameter to be alphanumeric.

--ERROR RH 60-- The NOS/BE permanent file specified has been dumped.

Condition Identifier: RHE\$FILE_DUMPED Code: RH 60

Description: The file you are trying to attach has been archived.

User Action: Contact operations personnel and have the file reloaded.

--ERROR RH 61-- NOS/BE has no APF space available.

Condition Identifier: RHE\$NO_APF_SPACE Code: RH 61

Description: All available entries in the Attached Permanent File (APF) table are in use.

User Action: Try again. If the condition persists, notify the system analyst.

--ERROR RH 62-- The NOS/BE file is attached elsewhere with exclusive access.

Condition Identifier: RHE\$PERMISSION_CONFLICTS Code: RH 62

Description: The file is attached to a job in the NOS/BE system with exclusive access.

User Action: Try again.

--ERROR RH 63-- The NOS/BE file's RBT chain is too large for the PFC.

Condition Identifier: RHE\$RBT_CHAIN_TOO_LARGE Code: RH 63

Description: The file is too large to fit in the permanent file catalog (PFC) table.

User Action: Make the file smaller before again trying to create a permanent file.

--ERROR RH 64-- The NOS/BE file specified resides on an unavailable device.

Condition Identifier: RHE\$UNAVAILABLE_DEVICE Code: RH 64

Description: The NOS/BE permanent file device is not mounted.

User Action: Notify the system analyst.

--ERROR RH 65-- The NOS/BE file specified is not available.

Condition Identifier: RHE\$FILE_NOT_AVAILABLE Code: RH 65

Description: The file is in an unusable state.

User Action: Notify analyst that the system requires a level 0 or 1 NOS/BE deadstart.

--ERROR RH 66-- The NOS/BE permanent file manager has been stopped by the system.

Condition Identifier: RHE\$PFM_STOPPED_BY_SYSTEM Code: RH 66

Description: The IRHF 170 partner job has been terminated abnormally without completing permanent file processing.

User Action: Notify the system analyst.

--ERROR RH 67-- The NOS/BE permission given is incorrect.

Condition Identifier: RHE\$INCORRECT_PERMISSION Code: RH 67

Description: You have attempted to get or replace an existing file with an incorrect password.

User Action: Correct the password and try again.

--ERROR RH 68-- NOS/BE had an I/O error processing this request.

Condition Identifier: RHE\$IO_ERROR Code: RH 68

Description: Read/write of PFC/PFD failed due to a disk error.

User Action: Try again. Notify the system analyst if the problem persists.

--ERROR RM 12-- Mass storage volume {text} is not a member of class {text}.

Condition Identifier: RME\$FILE_CLASS_NOT_VALID Code: RM 12

Description: The specified volume is not a member of the specified class.

User Action: Choose either a different volume or a different class.

--ERROR RM 15-- The ESTIMATED_FILE_SIZE capability of {text} is not implemented.

Condition Identifier: RME\$FILE_SIZE_NOT_IMPLEMENTED Code: RM 15

Description: The ESTIMATED_FILE_SIZE capability has not been implemented.

User Action: Remove the specification for ESTIMATED_FILE_SIZE.

--ERROR RM 21-- \$HIGH and \$LOW are improper cycle references on {text}.

Condition Identifier: RME\$IMPROPER_CYCLE_REFERENCE Code: RM 21

Description: \$HIGH and \$LOW are not allowed as part of the file reference because this request is creating the file and \$HIGH or \$LOW implies that the file exists.

User Action: Remove the \$HIGH or \$LOW specification.

--ERROR RM 24-- Volume: {text} is unknown.

Condition Identifier: RME\$UNKNOWN_VOLUME Code: RM 24

Description: The specified volume is unknown to the system.

User Action: Specify the correct volume.

--ERROR RM 25-- Access to requested volume is not allowed.

Condition Identifier: RME\$JOB_NOT_VALID Code: RM 25

Description: The requesting job does not have permission to perform the operation as requested.

User Action: Not applicable.

--ERROR RM 28-- Only SYSTEM and MAINTENANCE jobs are allowed to inhibit volume overflow.

Condition Identifier: RME\$VOLUME_OVERFLOW_REQUIRED Code: RM 28

Description: The user is not allowed to inhibit volume overflow.

User Action: Remove the specification for VOLUME_OVERFLOW_ALLOWED on the request.

--ERROR RM 35-- File {file} : DENSITY parameter of {text} is improper.

Condition Identifier: RME\$IMPROPER_DENSITY_VALUE Code: RM 35

Description: The value specified for the density parameter is invalid.

User Action: Correct the value by selecting the proper density.

--ERROR RM 40-- File {file} : {text} is an improper value for the RECORDED_VSN parameter of {text}.

Condition Identifier: RME\$IMPROPER_RECORDED_VSN_VALUE Code: RM 40

Description: One or more recorded volume serial numbers (VSN) in the array of volume descriptors was not well formed.

User Action: Correct the improperly recorded VSN.

--ERROR RM 45-- File {file} : {text} is an improper value for the EXTERNAL_VSN parameter of {text}.

Condition Identifier: RME\$IMPROPER_EXTERNAL_VSN_VALUE Code: RM 45

Description: One or more external volume serial numbers (VSN) in the array of volume descriptors was not well formed. Perhaps a string greater than six characters was given.

User Action: Correct the improper VSN.

--ERROR RM 50-- File {file} : WRITE RING parameter of {text} is improper.

Condition Identifier: RME\$IMPROPER_WRITE_RING_VALUE Code: RM 50

Description: A value other than a boolean value was provided.

User Action: Correct the value provided.

--ERROR RM 65-- File {file} : TERMINAL ATTRIBUTES parameter of {text} had improper KEY(S) in array element(s): {text}.

Condition Identifier: RME\$IMPROPER_TERM_ATTRIB_KEY Code: RM 65

Description: Each key field must contain a valid terminal attribute identifier.

User Action: Correct the value assigned to the key field.

--ERROR RM 70-- File {file} : TERMINAL ATTRIBUTES parameter of {text} had improper value(s) in array element(s): {text}.

Condition Identifier: RME\$IMPROPER_TERM_ATTRIB_VALUE Code: RM 70

Description: Each attribute field must contain a value valid for that attribute.

User Action: Refer to the attribute description to determine the valid ones and correct.

--WARNING RM 85-- File {file} is already assigned to a device of type {text}.

Condition Identifier: RME\$REDUNDANT_DEVICE_ASSIGNMENT Code: RM 85

Description: The file is already assigned to the requested device, therefore, there is no need to issue this request.

User Action: Do not issue this request for this file.

--ERROR RM 87-- File {file} : Current assignment to a {text} device conflicts with request for assignment to a {text} device.

Condition Identifier: RME\$DEVICE_ASSIGNMENT_CONFLICT Code: RM 87

Description: A file can only be assigned to one device at a time. The specified file is already assigned to a device that differs from the device specified on the request.

User Action: Either delete the file from the device to which it is currently assigned and reassign it to the requested device, or do not issue this request.

--INFORMATIVE RM 705-- the specified tape volume {text} is currently assigned

Condition Identifier: RME\$VOLUME_ALREADY_ASSIGNED Code: RM 705

Description: The TERMINATE_TAPE_ASSIGNMENT console command was issued and the volume (EVSN) identified is already assigned.

User Action: Correct the parameter or do not issue the command.

--INFORMATIVE RM 707-- no tape mount request pending for specified volume: {text}

Condition Identifier: RME\$NO_TAPE_MOUNT_PENDING Code: RM 707

Description: The ASSIGN_DEVICE or TERMINATE_TAPE_ASSIGNMENT command was issued for a volume (EVSN) that has no pending tape mount request.

User Action: Correct the parameter or do not issue the command.

--INFORMATIVE RM 708-- the system_name is required for unique assignment of volume: {text}

Condition Identifier: RME\$SYSTEM_NAME_REQUIRED Code: RM 708

Description: The ASSIGN_DEVICE command was issued for a volume (EVSN) that has 2 or more tape mount requests pending.

User Action: Reissue the command specifying the JOB_NAME parameter.

--INFORMATIVE RM 709-- the specified element_name of {text} is undefined

Condition Identifier: RME\$UNDEFINED_ELEMENT_NAME Code: RM 709

Description: The ASSIGN_DEVICE or REASSIGN_DEVICE command was issued with an element name that does not exist.

User Action: Reissue the command specifying a valid element name.

--INFORMATIVE RM 710-- the specified element_name of {text} is already assigned

Condition Identifier: RME\$ELEMENT_NAME_ASSIGNED Code: RM 710

Description: The ASSIGN_DEVICE command was issued specifying an element that is already assigned.

User Action: Reissue the command specifying the element name of an unassigned tape.

--INFORMATIVE RM 711-- the specified element_name of {text} is not ON

Condition Identifier: RME\$ELEMENT_NAME_NOT_ON Code: RM 711

Description: The ASSIGN_DEVICE or REASSIGN_DEVICE console command was issued to a tape element that is in an OFF or DOWN state.

User Action: Reissue the command specifying the correct element name.

--INFORMATIVE RM 712-- the specified element_name of {text} is not currently assigned

Condition Identifier: RME\$ELEMENT_NAME_NOT_ASSIGNED Code: RM 712

Description: The REASSIGN_DEVICE command was issued to a tape element that is not assigned.

User Action: Reissue the command with the correct element name.

--INFORMATIVE RM 713-- the system_name is required for unique termination of volume: {text}

Condition Identifier: RME\$SYSTEM_NAME_REQ_FOR_TERM Code: RM 713

Description: The TERMINATE_TAPE_ASSIGNMENT command was issued for a volume (EVSN) that is not unique.

User Action: Reissue the command specifying the JOB_NAME parameter.

--ERROR SC 10-- ALL not allowed for {text}.

Condition Identifier: SCE\$ALL_NOT_ALLOWED Code: SC 10

Description: The keyword ALL is not valid for this parameter.

User Action: Remove the keyword ALL and specify a valid parameter value.

--ERROR SC 20-- ALL is not allowed as a member of a list for {text}.

Condition Identifier: SCE\$ALL_USED_IN_LIST Code: SC 20

Description: The keyword ALL cannot be included within the list for this parameter.

User Action: Remove the keyword ALL from the list.

--ERROR SC 25-- ALL is not allowed as part of a range.

Condition Identifier: SCE\$ALL_USED_IN_RANGE Code: SC 25

Description: The keyword ALL is not valid as a range bound. The keywords that can be used as range bounds are FIRST, LAST, and CURRENT.

User Action: Replace the keyword ALL with a valid range bound.

--ERROR SC 30-- Allocation failed for {text}.

Condition Identifier: SCE\$ALLOCATION_FAILED Code: SC 30

Description: SCU could not allocate sufficient space for this structure.

User Action: Either request a larger space limit or split the operation into two or more smaller operations if possible.

--ERROR SC 33-- Command is not allowed after a base library has been established.

Condition Identifier: SCE\$CHANGED_BASE_LIBRARY Code: SC 33

Description: A different base library may not be selected by create_library or use_library after a base library has been established.

User Action: End the current session and start a new session to access a different base library.

--ERROR SC 34-- Allocation of space in the communication sequence has failed.

Condition Identifier: SCE\$COM_ALLOCATION_FAILED Code: SC 34

Description: Allocation of space in a communication sequence has failed.

User Action: Check the pointer to sequence being passed to the request.

--ERROR SC 35-- Combining altered modification {text} requires authority {text}.

Condition Identifier: SCE\$COMBINE_REQUIRES_AUTHORITY Code: SC 35

Description: A modification on a subset library has been changed more recently than on the base library.

Combining these libraries requires the listed authority on the base library.

User Action: Set authority on the base library to the required level.

--ERROR SC 40-- Command {text} not allowed in {text}.

Condition Identifier: SCE\$COMMAND_NOT_ALLOWED Code: SC 40

Description: This command is not valid within this processing mode. When reading editor or selection criteria commands, the SCL interpreter accepts only certain SCU commands. The SCU command description states if the command is valid in these other processing modes.

User Action: None.

--ERROR SC 45-- {text} requires authority {text}.

Condition Identifier: SCE\$COMMAND_REQUIRES_AUTHORITY Code: SC 45

Description: You do not have the authority required for the command as indicated in the application_info field of your file permit entry for the library file.

User Action: Request the appropriate authority from the owner of the file.

--WARNING SC 60-- Creation modification excluded for deck {text}.

Condition Identifier: SCE\$CREATION_MOD_EXCLUDED Code: SC 60

Description: Because the selection criteria for the command excluded the original modification specified when the deck was created, the deck was not selected.

User Action: If the deck is to be selected, do not exclude its creation modification.

--WARNING SC 65-- Creation modification for deck {text} not state 4.

Condition Identifier: SCE\$CREATION_MOD_NOT_STATE_4 Code: SC 65

Description: A deck may not be resequenced unless the creation modification is in state 4.

User Action: Change the creation modification for the deck to state 4.

--ERROR SC 70-- Selection criteria command {text} not valid on EXTRACT_SOURCE_LIBRARY with INTERLOCK selected.

Condition Identifier: SCE\$CRITERIA_ENTRY_PROHIBITED Code: SC 70

Description: This selection criteria command attempted to exclude a modification for a deck whose interlock field has been set. An interlocked deck must be extracted whole.

User Action: Either enter a command to exclude the interlocked deck or omit the command to exclude the modification.

--ERROR SC 75-- Cycle selector not found.

Condition Identifier: SCE\$CYCLE_SELECTOR_NOT_FOUND Code: SC 75

Description: In the process of examining a file reference in an output source map, no cycle selector followed a period.

User Action: None.

--ERROR SC 80-- Deck {text} already exists.

Condition Identifier: SCE\$DECK_ALREADY_EXISTS Code: SC 80

Description: The deck cannot be created because a deck with that name already exists in the library. Note that SCU does not distinguish between uppercase and lowercase letters in names.

User Action: Specify another name or delete the existing deck.

--WARNING SC 82-- Possible data loss from deck {text} extension.

Condition Identifier: SCE\$DECK_EXTENSION_WARNING Code: SC 82

Description: The program CONVERT_SCU10_TO_SCU11 was unable to correctly interpret all of a deck header extension.

User Action: None.

--WARNING SC 85-- Deck {text} interlocked for {text}.

Condition Identifier: SCE\$DECK_INTERLOCKED Code: SC 85

Description: The deck currently has the specified interlock. Any changes made to the deck while it is interlocked would be discarded in a new library creating by merging the library with the extracted library.

User Action: None.

--ERROR SC 90-- Deck limit exceeded.

Condition Identifier: SCE\$DECK_LIMIT_EXCEEDED Code: SC 90

Description: Creation of another deck would have exceeded the limit of 262,143 decks per library.

User Action: Split the library by extracting a set of decks to form a new library and deleting those decks from the existing library.

--ERROR SC 100-- Deck modification limit exceeded.

Condition Identifier: SCE\$DECK_MOD_LIMIT_EXCEEDED Code: SC 100

Description: Creation of another modification would have exceeded the limit of 16,383 modifications per deck.

User Action: If appropriate, sequence the deck. Sequencing a deck consolidates all modifications in state 4 into the creation modification.

--WARNING SC 105-- Deck {text} is not expandable.

Condition Identifier: SCE\$DECK_NOT_EXPANDABLE Code: SC 105

Description: The specified deck was not expanded because its expand attribute is false.

User Action: If the deck is to be expanded, issue a change_deck command to change the expand attribute to TRUE.

--WARNING SC 110-- Deck {text} not on base.

Condition Identifier: SCE\$DECK_NOT_ON_BASE Code: SC 110

Description: The working library does not contain a deck by this name.

User Action: Either correct the deck name or create a deck by that name.

--ERROR SC 120-- Deck {text} not referenced.

Condition Identifier: SCE\$DECK_NOT_REFERENCED Code: SC 120

Description: The command lists no references because no references to the deck exist.

User Action: Ensure that the deck name is specified correctly.

--WARNING SC 130-- Deck {text} duplicated on libraries {file} {file} {file} {file} {file} {file} {file} {file} {file} {file}

Condition Identifier: SCE\$DECK_ON_TWO_LIBRARIES Code: SC 130

Description: Because more than one source library contains a deck having the same name, none of the source library versions of the deck are copied to the working library. However, if a version of the deck already exists on the working library, it remains.

User Action: Determine if the desired version of the deck is within the working library.

--WARNING SC 135-- A text line ends with the chosen delimiter. See EDIT_ COMMANDS file {file}.

Condition Identifier: SCE\$DELIMITER_IN_NEW_TEXT Code: SC 135

Description: The file of editor commands and text generated is in error because at least one text line ends with the chosen terminating delimiter.

User Action: Either change the delimiter string both on the command and at the end of the inserted text or repeat the command generation specifying another terminating delimiter.

--ERROR SC 140-- Line {text} in deck {text} ends with the chosen delimiter.

Condition Identifier: SCE\$DELIMITER_TERMINATES_LINE Code: SC 140

Description: The sequence of editor commands and text extracted is in error because at least one text line ends with the chosen terminating delimiter.

User Action: Either change the delimiter string both on the command and at the end of the inserted text or extract the modification again specifying another terminating delimiter.

--ERROR SC 145-- Display information stack exceeded.

Condition Identifier: SCE\$DISPLAY_INFO_STACK_EXCEEDED Code: SC 145

Description: Internal SCU error. Information has been stacked during escapes from restricted command searching.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR SC 150-- Duplicate deck name {text}.

Condition Identifier: SCE\$DUPLICATE_DECK_NAME Code: SC 150

Description: The command specifies the same deck more than once.

User Action: Remove the redundant deck specification.

--ERROR SC 160-- Duplicate feature name {text}.

Condition Identifier: SCE\$DUPLICATE_FEATURE_NAME Code: SC 160

Description: The command specifies the same feature more than once.

User Action: Remove the redundant feature specification.

--ERROR SC 170-- Duplicate group name {text}.

Condition Identifier: SCE\$DUPLICATE_GROUP_NAME Code: SC 170

Description: The command specifies the same group more than once.

User Action: Remove the redundant feature specification.

--ERROR SC 190-- Duplicate modification name {text}.

Condition Identifier: SCE\$DUPLICATE_MOD_NAME Code: SC 190

Description: The command specifies the same modification more than once.

User Action: Remove the redundant modification specification.

--ERROR SC 200-- Errors closing libraries.

Condition Identifier: SCE\$ERRORS_CLOSING_LIBRARIES Code: SC 200

Description: SCU could not close the library files at completion of the command.

User Action: Preceding messages may explain why the files could not be closed.

--ERROR SC 210-- Errors converting library - {file}.

Condition Identifier: SCE\$ERRORS_CONVERTING_LIBRARY Code: SC 210

Description: Messages written to the listing file describe errors that occurred during the file conversion. If the listing file is the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 220-- Errors merging directories. See file {file}.

Condition Identifier: SCE\$ERRORS_MERGING_DIRECTORIES Code: SC 220

Description: Messages written to the \$ERRORS file describe errors that occurred during the library directory merging. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 230-- Errors merging modification lists. See file {file}.

Condition Identifier: SCE\$ERRORS_MERGING_MOD_LISTS Code: SC 230

Description: Messages written to the \$ERRORS file describe errors that occurred during the library modification list merging. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 240-- Errors processing command {text}. See file {file}.

Condition Identifier: SCE\$ERRORS_PROCESSING_COMMAND Code: SC 240

Description: Messages written to the \$ERRORS file describe errors that occurred during command processing. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 250-- Errors processing selection_criteria. See file {file}.

Condition Identifier: SCE\$ERRORS_PROCESSING_CRITERIA Code: SC 250

Description: Messages written to the \$ERRORS file describe errors that occurred during selection criteria processing. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 260-- Errors processing deck {text}.

Condition Identifier: SCE\$ERRORS_PROCESSING_DECK Code: SC 260

Description: The preceding error messages apply to this deck.

User Action: Correct the errors listed.

--ERROR SC 265-- Errors processing line at {text}.

Condition Identifier: SCE\$ERRORS_PROCESSING_LINE Code: SC 265

Description: The preceding error messages apply to this line.

User Action: Correct the errors listed.

--ERROR SC 268-- Errors processing name_list file - {file}. See LIST file.

Condition Identifier: SCE\$ERRORS_PROCESSING_NAME_LIST Code: SC 268

Description: Messages written to the listing file describe errors that occurred during processing of the file of name substitutions. If the listing file is the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 270-- Errors processing parameter {text}.

Condition Identifier: SCE\$ERRORS_PROCESSING_PARAM Code: SC 270

Description: Messages written to the \$ERRORS file describe errors that occurred during parameter processing. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: Correct the errors listed.

--ERROR SC 280-- Expecting deck directive. Found - {text}.

Condition Identifier: SCE\$EXPECTING_DECK_DIRECTIVE Code: SC 280

Description: Because you specified DECK_DIRECTIVES_INCLUDED=TRUE on the command, SCU expected the first line of source text read to be a DECK directive.

User Action: Ensure that the correct text file is specified on the command. If so, edit the file so that the first line is a DECK directive. The DECK directive must begin with the key character for the library.

--ERROR SC 290-- Feature {text} already exists.

Condition Identifier: SCE\$FEATURE_ALREADY_EXISTS Code: SC 290

Description: The feature cannot be created because a feature by that name already exists in the feature list.
Note that SCU does not distinguish between uppercase and lowercase letters in names.

User Action: Specify another name.

--ERROR SC 300-- Feature limit exceeded.

Condition Identifier: SCE\$FEATURE_LIMIT_EXCEEDED Code: SC 300

Description: Creation of another feature would have exceeded the limit of 65,535 features per library.

User Action: Either specify an existing feature name or generate a new library having fewer features. To do so, extract a new library containing only the desired features.

--ERROR SC 310-- File {file} used for conflicting purposes.

Condition Identifier: SCE\$FILE_NAME_CONFLICT Code: SC 310

Description: An attempt was made to use the same file for two purposes, such as, for use as both the base library and a list file on a command such as change_deck_names.

User Action: Specify two different files.

--ERROR SC 312-- A value must be given for either feature or modification.

Condition Identifier: SCE\$FEATURE_OR_MOD_REQUIRED Code: SC 312

Description: Either a feature or modification parameter must be specified.

User Action: Specify either a feature or modification and reissue the command.

--ERROR SC 314-- Library format version mismatch on file {file}.

Condition Identifier: SCE\$FORMAT_VERSION_MISMATCH Code: SC 314

Description: The library format version in the library header differs from the library format version in the working library header.

User Action: None.

--ERROR SC 320-- Group {text} already exists.

Condition Identifier: SCE\$GROUP_ALREADY_EXISTS Code: SC 320

Description: The group cannot be created because a group by that name already exists in the library. Note that SCU does not distinguish between uppercase and lowercase letters in names.

User Action: Specify another name.

--ERROR SC 330-- Group limit exceeded.

Condition Identifier: SCE\$GROUP_LIMIT_EXCEEDED Code: SC 330

Description: Creation of another group would have exceeded either the limit of 255 groups per deck or the limit of 32,767 groups per library.

User Action: Either specify an existing group name or generate a new library having fewer groups. To do so, extract a new library excluding one or more groups of decks.

--ERROR SC 350-- History limit exceeded for line {text}.

Condition Identifier: SCE\$HISTORY_LIMIT_EXCEEDED Code: SC 350

Description: Another modification of the line would have exceeded the limit of 255 modifications per line.

User Action: If appropriate, sequence the deck. Sequencing a deck consolidates all modifications in state 4 into the creation modification.

--ERROR SC 355-- the info sequence supplied by the user has overflowed.

Condition Identifier: SCE\$INFO_SEQUENCE_OVERFLOW Code: SC 355

Description: The sequence passed for this parameter is too small.

User Action: Check the pointer to sequence passed.

--ERROR SC 360-- Interlock violation for deck {text} on libraries {file} and {file}.

Condition Identifier: SCE\$INTERLOCK_VIOLATION Code: SC 360

Description: A deck on each of the libraries has the same name but the user name in the subinterlock field of the base copy and the user name in the original interlock field of the extracted copy do not match. The interlock was set when a user extracted a copy of the deck.

User Action: Ensure that the correct libraries are being merged. To ignore the interlock, specify INTERLOCK=FALSE on the command.

--WARNING SC 365-- Interlock violation for deck {text} on libraries {file} and {file}.

Condition Identifier: SCE\$INTERLOCK_VIOLATION_WARNING Code: SC 365

Description: A deck on each of the libraries has the same name but the user name in the subinterlock field of the base copy and the user name in the original interlock field of the extracted copy do not match. The interlock was set when a user extracted a copy of the deck.

User Action: Ensure that the correct libraries are being merged. To ignore the interlock, specify INTERLOCK=FALSE on the command.

--ERROR SC 370-- Internal error {text}.

Condition Identifier: SCE\$INTERNAL_ERROR Code: SC 370

Description: SCU internal error.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR SC 380-- Invalid copy directive {text} at {text}.

Condition Identifier: SCE\$INVALID_COPY_DIRECTIVE Code: SC 380

Description: The copy directive has incorrect syntax. It must begin with the key character followed by COPY or COPYC, spaces and/or a comma, and the deck name. It can also include SCL comments. The first substituted field gives the text of the directive; the second, the line identifier of the line on which it occurs.

User Action: Correct the directive.

--ERROR SC 390-- Invalid create index.

Condition Identifier: SCE\$INVALID_CREATE_INDEX Code: SC 390

Description: SCU internal error.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR SC 395-- Deck attribute key out of range for {text}

Condition Identifier: SCE\$INVALID_DECK_ATTR_KEY Code: SC 395

Description: One of the deck attribute keys given on the request is out of range.

User Action: None.

--ERROR SC 396-- Invalid deck name {text}.

Condition Identifier: SCE\$INVALID_DECK_NAME Code: SC 396

Description: The value specified for a deck name is not a valid SCL name.

User Action: None.

--ERROR SC 397-- Invalid feature name {text}.

Condition Identifier: SCE\$INVALID_FEATURE_NAME Code: SC 397

Description: The value given for a feature name is not a valid SCL name.

User Action: None.

--ERROR SC 398-- Invalid group name {text}.

Condition Identifier: SCE\$INVALID_GROUP_NAME Code: SC 398

Description: The value given for a group name is not a valid SCL name.

User Action: None.

--ERROR SC 399-- Invalid mod name {text}.

Condition Identifier: SCE\$INVALID_MOD_NAME Code: SC 399

Description: The value given for a modification name is not a valid SCL name.

User Action: Not applicable.

--ERROR SC 400-- Invalid history modification index.

Condition Identifier: SCE\$INVALID_HISTORY_MOD_INDEX Code: SC 400

Description: SCU internal error.

User Action: Notify a site analyst, providing appropriate documentation.

--ERROR SC 405-- Invalid keyword value.

Condition Identifier: SCE\$INVALID_KEYWORD_VALUE Code: SC 405

Description: The keyword value on the LINES or INSERTION_LOCATION parameter is invalid. The valid keyword values for both parameters are FIRST, LAST, and CURRENT; the keyword value ALL is also valid for the LINES parameter.

User Action: Correct the parameter value.

--ERROR SC 408-- Library attribute key out of range for {text}

Condition Identifier: SCE\$INVALID_LIBRARY_ATTR_KEY Code: SC 408

Description: The user has specified a library attribute key that is out of range.

User Action: None.

--ERROR SC 410-- Invalid library file {file}.

Condition Identifier: SCE\$INVALID_LIBRARY_FILE Code: SC 410

Description: When SCU opened the file, it determined that the file does not contain a source library in SCU format.

User Action: Ensure that the correct file is specified. To create an empty source library, enter the CREATE_SOURCE_LIBRARY command.

--ERROR SC 416-- Modification attribute key out of range for {text}

Condition Identifier: SCE\$INVALID_MOD_ATTR_KEY Code: SC 416

Description: The user has specified a modification attribute key that is out of range.

User Action: None.

--ERROR SC 418-- Invalid range of columns {text} to {text}.

Condition Identifier: SCE\$INVALID_RANGE_OF_COLUMNS Code: SC 418

Description: The integer specifying the lower bound of the range must be less than or equal to the integer specifying the upper bound of the range.

User Action: Correct the column range specification.

--ERROR SC 420-- Invalid range of decks {text} to {text}.

Condition Identifier: SCE\$INVALID_RANGE_OF_DECKS Code: SC 420

Description: The last deck in the range must follow the first deck in the range within the library deck list. The deck list is alphabetical.

User Action: Correct the deck range specification.

--ERROR SC 430-- Invalid range of features {text} to {text}.

Condition Identifier: SCE\$INVALID_RANGE_OF_FEATURES Code: SC 430

Description: The last feature in the range must follow the first feature in the range within the library feature list.

User Action: Correct the feature range specification. To display the feature list order, enter DISPLAY_FEATURE_LIST.

--ERROR SC 440-- Invalid range of groups {text} to {text}.

Condition Identifier: SCE\$INVALID_RANGE_OF_GROUPS Code: SC 440

Description: The last group in the range must follow the first group in the range within the library group list.

User Action: Correct the group range specification. To display the group list order, enter DISPLAY_GROUP_LIST.

--ERROR SC 450-- Invalid range of modifications {text} to {text}.

Condition Identifier: SCE\$INVALID_RANGE_OF_MODS Code: SC 450

Description: The last modification in the range must follow the first modification in the range within the library modification list.

User Action: Correct the modification range specification. To display the modification list order, enter DISPLAY_MODIFICATION_LIST.

--ERROR SC 460-- Key character mismatch.

Condition Identifier: SCE\$KEY_CHARACTER_MISMATCH Code: SC 460

Description: The libraries cannot be merged because they use different key characters.

User Action: To display the key character of a library header, enter DISPLAY_LIBRARY. To merge the contents of the libraries, you must extract the deck text from one of the libraries, edit the text so all directives begin with the correct key character, and then create decks on another library using the extracted text.

--ERROR SC 470-- Last deck on library.

Condition Identifier: SCE\$LAST_DECK_ON_LIBRARY Code: SC 470

Description: The NEXT_DECK function failed because the current deck is the last deck on the library.

User Action: Within an editor session, you can determine if the current deck is the last deck by comparing the values returned by the \$CURRENT_DECK and \$LAST_DECK functions.

--ERROR SC 480-- Last modification on library.

Condition Identifier: SCE\$LAST_MOD_ON_LIBRARY Code: SC 480

Description: The NEXT_MODIFICATION function failed because the current modification is the last modification in the modification list.

User Action: You can determine if the current modification is the last modification using the \$LAST_MODIFICATION function.

--WARNING SC 485-- Possible data loss from library {file} extension.

Condition Identifier: SCE\$LIBRARY_EXTENSION_WARNING Code: SC 485

Description: The program CONVERT_SCU10_TO_SCU11 was unable to correctly interpret all of a library header extension.

User Action: None.

--ERROR SC 490-- Library modification limit exceeded.

Condition Identifier: SCE\$LIBRARY_MOD_LIMIT_EXCEEDED Code: SC 490

Description: Creation of another modification would have exceeded the limit of 262,143 modifications per library.

User Action: If appropriate, sequence the library decks. Sequencing decks eliminates all modifications in state 4 except the creation modification.

--INFORMATIVE SC 500-- The library in the SDT was {file}. The library used was {file}.

Condition Identifier: SCE\$LIBRARY_USED Code: SC 500

Description: This message supplements sce\$regenerate_source_aborted, which specified which available cycles of the source libraries were actually used, as opposed to those recorded in the supplemental debug table in the object library being used.

User Action: None.

--ERROR SC 510-- Line length exceeded.

Condition Identifier: SCE\$LINE_LENGTH_EXCEEDED Code: SC 510

Description: The line would have exceeded the maximum line length of 256 characters.

User Action: Enter the line as two lines.

--ERROR SC 520-- Line limit exceeded.

Condition Identifier: SCE\$LINE_LIMIT_EXCEEDED Code: SC 520

Description: Addition of the line would have exceeded the limit of 16,777,215 lines per deck.

User Action: Create two decks using the text of the deck. To do so, insert a DECK directive at the point where the deck should be split, extract the deck text, and then create the decks using the extracted text.

--ERROR SC 560-- The first encoded line in a supplemental debug table has no origin index.

Condition Identifier: SCE\$LINE_ORIGIN_NOT_GIVEN Code: SC 560

Description: The SCP\$REGENERATE_SOURCE procedure has been unable to access the origin index for an encoded line in a loader table.

User Action: The supplemental debug table will probably have to be regenerated.

--ERROR SC 580-- Line truncated at {text}.

Condition Identifier: SCE\$LINE_TRUNCATED Code: SC 580

Description: The line is truncated because it is longer than the current line width setting.

User Action: Either increase the line width setting or split the line.

--ERROR SC 605-- Missing deck name on *DECK directive.

Condition Identifier: SCE\$MISSING_DECK_NAME Code: SC 605

Description: The required deck name is missing from a DECK directive.

User Action: Add the deck name to the DECK directive and rerun.

--ERROR SC 610-- Missing {text} directive.

Condition Identifier: SCE\$MISSING_DIRECTIVE Code: SC 610

Description: While expanding text, SCU processed a directive that began a block, but it did not find a matching directive to end the block. A BLOCK must have a matching BLOCKEND, an IF must have a matching IFEND, and a TEXT must have a matching TEXTEND directive.

User Action: Insert the required directive at the appropriate location in the text.

--ERROR SC 620-- Modification {text} already exists.

Condition Identifier: SCE\$MOD_ALREADY_EXISTS Code: SC 620

Description: The modification cannot be created because a modification with that name already exists in the modification list.

User Action: Either continue the existing modification or specify another name for the new modification (uppercase/lowercase).

--ERROR SC 630-- Creation dates for modification {text} disagree.

Condition Identifier: SCE\$MOD_CREATION_DATE_MISMATCH Code: SC 630

Description: Although both libraries contain a modification with the same name, the creation dates in the modification headers differ. The modification was created within each library on a different date.

User Action: If desired, display the modification header to determine the date used.

--ERROR SC 640-- Creation times for modification {text} disagree.

Condition Identifier: SCE\$MOD_CREATION_TIME_MISMATCH Code: SC 640

Description: Although both libraries contain a modification with the same name, the creation times in the modification headers differ. The modification was created within each library at a different time.

User Action: If desired, display the modification header to determine the time used.

--WARNING SC 642-- Creation times for modification {text} disagree.

Condition Identifier: SCE\$MOD_CREATION_TIME_WARNING Code: SC 642

Description: Although both libraries contain a modification with the same name, the creation times in the modification headers differ. The modification was created within each library at a different time.

User Action: None.

--WARNING SC 645-- Possible data loss from modification {text} extension.

Condition Identifier: SCE\$MOD_EXTENSION_WARNING Code: SC 645

Description: The program CONVERT_SCU10_TO_SCU11 was unable to correctly interpret all of a modification header extension.

User Action: None.

--WARNING SC 647-- Modification {text} in maximum state and cannot be changed.

Condition Identifier: SCE\$MOD_MAX_STATE Code: SC 647

Description: Having been assigned state 4, the modification state and the modification itself cannot be changed.

User Action: Enter further changes to deck text using either a new modification or an existing modification not in state 4.

--ERROR SC 649-- Modification {text} is not the deck mod list of deck {text}.

Condition Identifier: SCE\$MOD_NOT_MAPPED_IN_DECK Code: SC 649

Description: During an attempt to regenerate source, an encoded line has been found that was introduced by a modification that no longer applies to this deck.

User Action: The source will probably need to be reexpanded from the original libraries or files.

--ERROR SC 650-- Modification {text} is not state 0.

Condition Identifier: SCE\$MOD_NOT_STATE_ZERO Code: SC 650

Description: The modification cannot be continued because it is not in state 0.

User Action: Either lower the modification state (if you have the appropriate authority for the file) or create a new modification.

--ERROR SC 655-- Modification {text} is not state 4.

Condition Identifier: SCE\$MOD_NOT_STATE_4 Code: SC 655

Description: Some operations such as sequence_deck require that the associated modification be at state 4. In this case it is not.

User Action: Not applicable.

--ERROR SC 660-- Multiple deck name sources.

Condition Identifier: SCE\$MULTIPLE_DECK_NAME_SOURCES Code: SC 660

Description: The command specified both the DECK parameter and DECK_DIRECTIVES_INCLUDED=TRUE.

User Action: Omit either the DECK parameter or the DECK_DIRECTIVES_INCLUDED parameter.

--ERROR SC 680-- No alternate bases specified.

Condition Identifier: SCE\$NO_ALT_BASES_SPECIFIED Code: SC 680

Description: An EXCLUDE_LIBRARY selection criteria command attempted to exclude a library for an SCU command that did not specify the ALTERNATE_BASES parameter.

User Action: Either omit the EXCLUDE_LIBRARY command or specify the ALTERNATE_BASES parameter on the SCU command.

--WARNING SC 685-- No changes specified. No action taken.

Condition Identifier: SCE\$NO_CHANGES_SPECIFIED Code: SC 685

Description: The command did not specify any header changes.

User Action: If header changes are desired, specify the appropriate parameters to change header fields.

--ERROR SC 690-- No decks on library.

Condition Identifier: SCE\$NO_DECKS_ON_LIBRARY Code: SC 690

Description: The library has no decks.

User Action: If appropriate, create decks or add decks from another library.

--ERROR SC 700-- No decks selected.

Condition Identifier: SCE\$NO_DECKS_SELECTED Code: SC 700

Description: No decks met all selection criteria.

User Action: If appropriate, change the selection criteria command sequence.

--ERROR SC 710-- Either DECK or DECK_DIRECTIVES_INCLUDED parameter must be specified.

Condition Identifier: SCE\$NO_DECK_NAME_SPECIFIED Code: SC 710

Description: To create decks, a parameter must specify the source of the deck names. If DECK is specified, the names are on the parameter. If DECK_DIRECTIVES_INCLUDED=TRUE is specified, the deck names are on DECK directives in the text.

User Action: Specify DECK_DIRECTIVES_INCLUDED=TRUE if the source file text contains DECK directives; otherwise, specify the DECK parameter.

--ERROR SC 723-- No deck specified.

Condition Identifier: SCE\$NO_DECK_SPECIFIED Code: SC 723

Description: Self-explanatory.

User Action: None.

--ERROR SC 724-- No decks deleted.

Condition Identifier: SCE\$NO_DECKS_DELETED Code: SC 724

Description: All decks were skipped because of an error with each.

User Action: No user action required.

--ERROR SC 725-- No modification specified.

Condition Identifier: SCE\$NO_MODIFICATION_SPECIFIED Code: SC 725

Description: Self-explanatory.

User Action: None.

--ERROR SC 727-- No modifications deleted.

Condition Identifier: SCE\$NO_MODS_DELETED Code: SC 727

Description: All modifications were skipped because of some error with each.

User Action: No user action required.

--ERROR SC 730-- No modifications on library.

Condition Identifier: SCE\$NO_MODS_ON_LIBRARY Code: SC 730

Description: The library has no modifications.

User Action: To create a modification, use either the CREATE_MODIFICATION or CREATE_DECK commands.

--ERROR SC 735-- No decks combined.

Condition Identifier: SCE\$NO_OBJECTS_COMBINED Code: SC 735

Description: Either there were no objects to combine or the objects were all skipped due to an error condition.

User Action: None.

--WARNING SC 740-- No references on library.

Condition Identifier: SCE\$NO_REFERENCES_ON_LIBRARY Code: SC 740

Description: The library contains no copy directives that perform the type of reference specified on the command.

User Action: None.

--WARNING SC 745-- No SOURCE parameter specified, which is required when ddi=TRUE.

Condition Identifier: SCE\$NO_SOURCE_SPECIFIED Code: SC 745

Description: The user has indicated that deck names are specified by embedded directives in a source file but no source file was specified.

User Action: Retry the CREATE_DECK command with deck names specified by the deck parameter on the command or specify a source file.

--ERROR SC 751-- NONE not allowed for {text}.

Condition Identifier: SCE\$NONE_NOT_ALLOWED Code: SC 751

Description: The keyword NONE is not valid for this parameter.

User Action: Remove the keyword NONE and specify a valid parameter value.

--ERROR SC 752-- NONE is not allowed as a member of a list for {text}.

Condition Identifier: SCE\$NONE_USED_IN_LIST Code: SC 752

Description: The keyword NONE cannot be included within a list for this parameter.

User Action: Remove the keyword NONE from the list.

--ERROR SC 753-- NONE is not allowed as part of a range.

Condition Identifier: SCE\$NONE_USED_IN_RANGE Code: SC 753

Description: The keyword NONE is not valid as a range bound.

User Action: Replace the keyword NONE with a valid range bound.

--WARNING SC 754-- Not permitted to attach result file {file} in required access modes. Result file set to \$NULL.

Condition Identifier: SCE\$NOT_PERMITTED_TO_RESULT Code: SC 754

Description: The user is not permitted to attach the result file in the required access modes READ, SHORTEN and APPEND. The default result file has been set to \$NULL.

User Action: To save changes, write_library to an appropriate file.

--ERROR SC 755-- {text} of {text} requires authority {text}.

Condition Identifier: SCE\$OBJECT_REQUIRES_AUTHORITY Code: SC 755

Description: To perform the specified operation, you must have the appropriate authority granted in your file permit entry for the file.

User Action: Request the file owner for the appropriate authority.

--ERROR SC 760-- The origin table is empty.

Condition Identifier: SCE\$ORIGIN_TABLE_EMPTY Code: SC 760

Description: The origin table being accessed to regenerate source is empty.

User Action: The source for the deck or file will have to be obtained from the original library or file.

--ERROR SC 765-- Conflict in values for parameters {text} and {text}.

Condition Identifier: SCE\$PARAMETER_CONFLICT Code: SC 765

Description: On the DISPLAY_DECK_REFERENCES command, DECK_RESIDENCE=EXTERNAL conflicts with REFERENCE_DIRECTION=FROM.

User Action: Change either parameter value.

--ERROR SC 770-- Recursive copy at line {text} of deck {text}.

Condition Identifier: SCE\$RECURSIVE_COPY Code: SC 770

Description: The copy directive specified a deck already in the nested copy sequence, that is, a deck attempted to copy itself.

User Action: Either remove the copy directive that attempts to copy the deck already copied within the sequence or specify the EXPANSION_DEPTH parameter to limit the number of nesting levels expanded.

--FATAL SC 771-- Regenerate source aborted, item(s) in the Supplemental Debug Table were not found on any library of the libraries referenced.

Condition Identifier: SCE\$REGENERATE_SOURCE_ABORTED Code: SC 771

Description: A deck referenced in the supplemental debug table is not present on any of the libraries referenced. A series of sce\$library_used messages preceded this message detailing which cycle of each library was used.

User Action: Reproduce source from the object library by recompiling the module in question with the appropriate specification of source maps or by making accessible the source libraries itemized in the supplemental debug table.

--ERROR SC 773-- A write_library command is necessary to make the generated output source map valid. Let the result parameter default to the currently scheduled file. For possible additional errors see file {file}.

Condition Identifier: SCE\$RESULT_LIBRARY_IS_STALE Code: SC 773

Description: This message indicates that a deck whose only current text is on the working library has been encountered. The currently scheduled result library has been logged in the output source map as the origin for such decks.

User Action: The user can make the output source map correct by using the WRITE_LIBRARY command with the RESULT parameter omitted to write all decks to the currently scheduled result file before making additional changes.

--ERROR SC 775-- Result library written on file {file}.

Condition Identifier: SCE\$RESULT_LIBRARY_WRITTEN Code: SC 775

Description: Errors were encountered while trying to write the result library. The library has been written to the listed file.

User Action: Copy the listed file to an appropriate file.

--WARNING SC 785-- STATEMENT_IDENTIFIER attribute for file {file} cannot be reset.

Condition Identifier: SCE\$SI_ATTRIBUTE_MISMATCH Code: SC 785

Description: The file system currently does not allow the statement identifier attribute of a file to be changed to any other value once it has been initialized.

User Action: If this file is to be used by a process that is dependent on the statement identifier attribute, the data should be expanded onto a new file.

--ERROR SC 792-- The number of source files and deck names given disagree.

Condition Identifier: SCE\$SOURCE_NAMES_MISMATCH Code: SC 792

Description: The number of deck names specified is not the same as the number of source files specified.

User Action: Specify the same number of deck names and source files. To create an empty deck, specify \$NULL as its source file.

--FATAL SC 794-- The only current text for deck {text} is on the working library. A source library must be written to a file with the write_library (or end) command and the expand_deck repeated to generate a valid output source map.

Condition Identifier: SCE\$TEXT_ON_WORKING_LIBRARY Code: SC 794

Description: A deck whose only current text is on the working library has been encountered and the result library is currently scheduled to be written to \$NULL.

User Action: UTo make it possible for another EXPAND_DECK command to succeed in writing an output source map, the user must use the WRITE_LIBRARY command with the RESULT parameter specified to write all decks to a physical source library file.

--ERROR SC 795-- The text sequence supplied by the user has overflowed.

Condition Identifier: SCE\$TEXT_SEQUENCE_OVERFLOW Code: SC 795

Description: The sequence provided by the user was too small.

User Action: None.

--WARNING SC 797-- Time stamp on deck {text} differs from loader table.

Condition Identifier: SCE\$TIME_STAMPS_DIFFER_FOR_DECK Code: SC 797

Description: The origin table being accessed to regenerate the source for a deck indicates that it has been modified since the origin table was written.

User Action: The deck indicated should be recompiled to allow for reliable use of screen mode debugging.

--WARNING SC 798-- Time stamp on file {file} differs from loader table.

Condition Identifier: SCE\$TIME_STAMPS_DIFFER_FOR_FILE Code: SC 798

Description: A file has been modified since the object library in question was generated.

User Action: Re-expand and recompile this module so the program can be reliably debugged in screen mode.

--ERROR SC 800-- Too many alternate bases specified on request {text}.

Condition Identifier: SCE\$TOO_MANY_ALTERNATE_BASES Code: SC 800

Description: Self-explanatory.

User Action: A maximum of 255 alternate bases should be specified.

--ERROR SC 806-- Too many BLOCK levels {text}.

Condition Identifier: SCE\$TOO_MANY_BLOCK_LEVELS Code: SC 806

Description: Expansion of the BLOCK directive would exceed the maximum nesting levels of 32.

User Action: Reorganize the logic of the directives so that fewer nesting levels are required.

--ERROR SC 810-- Too many cross references.

Condition Identifier: SCE\$TOO_MANY_CROSS_REFS Code: SC 810

Description: The listing exceeded the cross reference entries limit of 4,193,303.

User Action: Limit the cross references listed using parameters on the command.

--ERROR SC 820-- Too many IF levels {text}.

Condition Identifier: SCE\$TOO_MANY_IF_LEVELS Code: SC 820

Description: Expansion of the IF directive would exceed the maximum nesting levels of 32.

User Action: Reorganize the logic of the directives so that fewer nesting levels are required.

--ERROR SC 825-- Objects of type {text} cannot be selected through the object list.

Condition Identifier: SCE\$TYPE_NOT_EXTRACTED Code: SC 825

Description: Only decks may currently be specified as objects to be extracted.

User Action: None.

--ERROR SC 830-- Unexpected directive {text} at {text}.

Condition Identifier: SCE\$UNEXPECTED_DIRECTIVE Code: SC 830

Description: One of the following was found.

- An IFEND, ELSE, or ELSEIF directive before an IF directive to begin the IF/IFEND block.
- A TEXTEND directive before a TEXT directive.
- A BLOCKEND directive before a BLOCK directive.
- More than one ELSE directive within an IF/IFEND block.
- An ELSEIF directive after the ELSE directive within an IF/IFEND block.

User Action: Correct the directive order.

--ERROR SC 840-- Unknown deck name {text} at {text}.

Condition Identifier: SCE\$UNKNOWN_DECK_NAME Code: SC 840

Description: The specified deck name is not in the deck list of the working library.

User Action: Ensure that the deck name is specified correctly.

--ERROR SC 850-- Unknown feature name {text}.

Condition Identifier: SCE\$UNKNOWN_FEATURE_NAME Code: SC 850

Description: The specified feature name is not in the feature list of the working library.

User Action: Ensure that the feature name is specified correctly.

--ERROR SC 860-- Unknown group name {text}.

Condition Identifier: SCE\$UNKNOWN_GROUP_NAME Code: SC 860

Description: The specified group name is not in the group list of the working library.

User Action: Ensure that the group name is specified correctly.

--ERROR SC 870-- Unknown library {text}.

Condition Identifier: SCE\$UNKNOWN_LIBRARY_NAME Code: SC 870

Description: The library name specified on the EXCLUDE_LIBRARY command was not specified on the ALTERNATE_BASES parameter of the command that requested selection criteria processing.

User Action: Either correct the library name on the EXCLUDE_LIBRARY command or specify the library on the ALTERNATE_BASES parameter.

--ERROR SC 880-- Unknown modification name {text}.

Condition Identifier: SCE\$UNKNOWN_MOD_NAME Code: SC 880

Description: The specified modification name is not in the modification list of the working library.

User Action: If the modification is an existing modification, ensure that its name is specified correctly. If the modification is to be a new modification, specify CONTINUE=FALSE on the command.

--WARNING SC 881-- Warning merging modification lists.

Condition Identifier: SCE\$WARNING_MERGING_MOD_LISTS Code: SC 881

Description: Messages written to the \$ERRORS file describe warning errors that occurred during library modification list merging. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: If appropriate, correct the errors listed.

--WARNING SC 882-- Headers differ for modification {text}.

Condition Identifier: SCE\$WARNING_MOD_HEADERS_DIFFER Code: SC 882

Description: A modification on the libraries has the same name but different information in the modification headers. The modification header from the base library will appear on the result file with the most recent time and date of modification from the two.

User Action: If desired, display the modification header.

--WARNING SC 883-- Warning merging directories.

Condition Identifier: SCE\$WARNING_MERGING_DIRECTORIES Code: SC 883

Description: Messages written to the \$ERRORS file describe warning errors that occurred during the library directory merging. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: If appropriate, correct the errors listed.

--WARNING SC 885-- Warnings processing command {text}. See file {file}.

Condition Identifier: SCE\$WARNINGS_DURING_COMMAND Code: SC 885

Description: Messages written to the \$ERRORS file describe warning errors that occurred during command processing. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: If appropriate, correct the errors listed.

--WARNING SC 888-- Warnings processing deck {text}.

Condition Identifier: SCE\$WARNINGS_DURING_DECK Code: SC 888

Description: Messages written to the \$ERRORS file describe warning errors that occurred during deck processing. If the \$ERRORS file is connected to the output file, the messages were displayed previously.

User Action: If appropriate, correct the errors listed.

--WARNING SC 890-- Warnings processing command {text} including interlock violations. See file {file}.

Condition Identifier: SCE\$WARNINGS_INCLUDE_INTERLOCK Code: SC 890

Description: Warnings were encountered in the processing of the command. At least one of them was an interlock command.

User Action: For the interactive user, this message is informative. It is intended primarily for Systems Command Language procedure writers.

--WARNING SC 895-- Deck {text} is not on any library used in the regenerate source process.

Condition Identifier: SCE\$WARNING_UNKNOWN_DECK_NAME Code: SC 895

Description: The specified deck name is not in the deck list of the working library.

User Action: Ensure that the deck name has been specified correctly.

--ERROR SF 20-- The job was recovered while reading the system routing control table, call {text} again.

Condition Identifier: SF\$CALL_AGAIN_JOB_RECOVERED Code: SF 20

Description: The job was recovered at a point where the specified program interface (typically SFP\$GET_ACTIVE_SYSTEM_STATS) was attempting to read the system routing control information for statistics. The program interface should be called again to allow the operation to complete.

User Action: Alter the code to call the specified interface again if this error is returned.

--ERROR SF 25-- Statistic code {text} counter array size must be less than {text}. ({text} request)

Condition Identifier: SF\$COUNTER_ARRAY_SIZE_RANGE Code: SF 25

Description: The array of counters passed to SFP\$EMIT_STATISTIC is too large.

User Action: Reduce the size of the array of counters to less than the number specified in the error message.

--ERROR SF 40-- Statistic code {text} descriptive data size must be less than {text}. ({text} request)

Condition Identifier: SF\$DESCRIPTIVE_DATA_SIZE Code: SF 40

Description: The string passed as the descriptive data parameter to SFP\$EMIT_STATISTIC was too long. The maximum size for the string is included in the diagnostic.

User Action: The code calling SFP\$EMIT_STATISTIC must be changed to reduce the size of the descriptive data to an acceptable length.

--ERROR SF 46-- User is not authorized to {text} the specified {text} routing control information.

Condition Identifier: SFE\$INSUFFICIENT_PRIVILEGE Code: SF 46

Description: The user is attempting to access statistic routing control information for a global log or the job account log. Only a system administrator (i.e. \$SYSTEM) can access statistic routing control information for these logs.

User Action: Do not attempt to retrieve or alter statistic routing control information for the global logs or the job account log.

--ERROR SF 47-- {text} is not a valid statistic name.

Condition Identifier: SFE\$INVALID_STATISTIC_NAME Code: SF 47

Description: The specified name does not conform to the specifications of a statistic name. A statistic name has the form XXNNNNNNNN, where XX is a 2 character ID and NNNNNNNN is a number in the range 00000000 through 16,777,215.

User Action: Change the statistic name.

--ERROR SF 90-- The work area passed to {text} is full.

Condition Identifier: SFE\$WORK_AREA_FULL Code: SF 90

Description: The work area passed to SFP\$GET_ACTIVE_JOB_STATISTICS or SFP\$GET_ACTIVE_SYSTEM_STATS is full.

User Action: Provide a larger work area.

--ERROR SF 1053-- The specified pointer to the array of limit records must not be a NIL pointer.

Condition Identifier: SFE\$LIMIT_ARRAY_POINTER_NIL Code: SF 1053

Description: The pointer to the limit array passed to SFP\$GET_ALL_JOB_LIMITS was NIL.

User Action: Pass a pointer to an array that will be large enough to hold all of the limits. SFP\$GET_JOB_LIMIT_COUNT can be used to determine the correct size.

--ERROR SF 1055-- The size of the limit array ({text} entries) must be large enough to hold all {text} limits.

Condition Identifier: SFE\$LIMIT_ARRAY_TOO_SMALL Code: SF 1055

Description: The limit array passed to SFP\$GET_ALL_JOB_LIMITS was not large enough to hold all of the job's limits.

User Action: Increase the size of the array. SFP\$GET_JOB_LIMIT_COUNT can be used to determine the required size.

--ERROR SF 1060-- {text} limit has not been activated.

Condition Identifier: SFE\$LIMIT_NOT_ACTIVATED Code: SF 1060

Description: A limit with the specified name is not active in the job.

User Action: Supply a correct limit name.

--WARNING SF 1065-- There are no limits activated for this job.

Condition Identifier: SFE\$NO_ACTIVE_LIMITS Code: SF 1065

Description: SFP\$GET_ALL_JOB_LIMITS was called by the system job (which is not limited).

User Action: Do not call SFP\$GET_ALL_JOB_LIMITS from the system job.

--ERROR SF 1085-- {text} is not a known job resource condition identifier.

Condition Identifier: SFE\$UNKNOWN_CONDITION_ID Code: SF 1085

Description: The condition identifier passed to SFP\$GET_JOB_LIMIT_NAME is not a condition identifier that has been associated with a limit.

User Action: Pass a correct condition identifier.

--FATAL SM 1-- SORT/MERGE user error; task not completed.

Condition Identifier: SME\$CONTROL_STATEMENT_ERROR Code: SM 1

Description: The sort or merge did not complete due to errors. Messages describing the errors were written to the errors file. If you did not specify an errors file, the default file \$ERRORS was used. However, if the \$ERRORS file was connected to the \$NULL file, the error messages were discarded.

User Action: Check the error file and correct the errors listed. If the error messages were discarded, reattempt the sort or merge. Remember to either connect the \$ERRORS file to a file or specify an errors file for the sort or merge.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the error parameter. See the FORTRAN Language Definition Usage manual for information on the SM5E procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$ERROR_FILE procedure. The CREATE_FILE_CONNECTION command description is in the NOS/VE System Usage manual.

--INFORMATIVE SM 2-- The first value in the RESULT_ARRAY was less than 0 or greater than 17 and was treated as 0 or 17 respectively.

Condition Identifier: SME\$BAD_RESULT_ARRAY_LENGTH Code: SM 2

Description: If you set the first element in the result array to a value less than zero, the value is treated as zero. If you set the first element to a value greater than 15, the value is treated as 15.

A value of 0 requests that no statistics be returned in the result array. A value of 15 returns statistics in elements 2 through 16 of the result array.

User Action: None required.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on result arrays.

--FATAL SM 3-- This SUM or KEY parameter ({text}) must be an integer.

Condition Identifier: SME\$INTEGER_FOR_POS_LENGTH Code: SM 3

Description: The values you specify for the position and length in a key field definition or the position, length, and repeat count in a sum field definition must be integer values.

User Action: Correct the sum or key field definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on key and sum parameters. See the FORTRAN Language Definition Usage manual for information on the SM5KEY and SM5SUM procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY and SMP\$SUM procedures.

--FATAL SM 4-- Too many values in ({text}).

Condition Identifier: SME\$TOO_MANY_PARAM_VALUE Code: SM 4

Description: Only four values can be specified in the value set for a sum or key field definition. The first three values specify the position, length and type of the field. The fourth value specifies the sort direction for a key field or the repeat count for a sum field.

User Action: Remove the extra values in the sum or key field definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on sum or key field definition. See the FORTRAN Language Definition Usage manual for information on the SM5KEY and SM5SUM procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY and SMP\$SUM procedures.

--FATAL SM 5-- A or D required as KEY order.

Condition Identifier: SME\$KEY_ORDER_VALUE Code: SM 5

Description: The only valid values for the sort order are A (for ascending) or D (for descending).

User Action: Specify either A or D for as the order value in a key field definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY parameter format. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure.

--FATAL SM 6-- ({text}) is not a KEY type name.

Condition Identifier: SME\$KEY_TYPE_NAME_REQUIRED Code: SM 6

Description: The key type you specified is not a predefined or user-defined collating sequence or a numeric format.

User Action: Correct the key type name. The predefined collating sequences are: ASCII, ASCII6, COBOL6, DISPLAY, EBCDIC, and EBCDIC6. The numeric data formats are: BINARY, BINARY_BITS, INTEGER, INTEGER_BITS, NUMERIC_FS, NUMERIC_LO, NUMERIC_LS, NUMERIC_NS, NUMERIC_TO, NUMERIC_TS, PACKED, PACKED_NS, and REAL.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on key types and the key parameter. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See The CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure.

--FATAL SM 7-- Duplicate parameter specified:({text}).

Condition Identifier: SME\$DUPLICATE_PARAMETER Code: SM 7

Description: A Sort/Merge parameter was specified more than once on a SORT or MERGE command or directive.

User Action: Remove the duplicate parameter.

Further Information: See the NOS/VE Advanced File Management Usage manual for information the Sort/Merge Parameters.

--FATAL SM 8-- Incorrect range in ESTIMATED_NUMBER_RECORDS parameter.

Condition Identifier: SME\$ENR_RANGE_INCORRECT Code: SM 8

Description: If you specify an estimated number of records range, the range must not extend beyond the valid range 1 through 16777215.

User Action: Correct the specified range or remove the parameter. NOS/VE Sort/Merge does not use the ENR value.

--WARNING SM 9-- SM5FAST is ignored.

Condition Identifier: SME\$FASTIO_IGNORED Code: SM 9

Description: The parameter FASTIO and procedure SM5FAST are available only when using NOS Sort/Merge Version 5.

User Action: None required.

--FATAL SM 10-- The definition of a collating sequence must also specify COLLATING_SEQUENCE_STEP characters or COLLATING_SEQUENCE_REMAINDER.

Condition Identifier: SME\$INCOMPLETE_SEQN Code: SM 10

Description: A collating sequence definition must specify at least one value step.

User Action: Put a value step definition parameter or call immediately after the collating sequence name parameter or call.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the collating sequence parameters. See the FORTRAN Language Definition Usage manual for information on SM5SEQX procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_X procedures.

--FATAL SM 11-- Your COLLATING_SEQUENCE_NAME uses a predefined KEY type name ({text}).

Condition Identifier: SME\$INCORRECT_SEQN_NAME Code: SM 11

Description: The user-defined collating sequence name cannot be the same as that of a predefined collating sequence name.

User Action: Check your collating sequence name. The predefined names are: ASCII, ASCII6, COBOL6, DISPLAY, EBCDIC, and EBCDIC6.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the predefined collating sequence names. See the FORTRAN Language Definition Usage manual for information on the SM5SEQX procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_X procedures.

--FATAL SM 12-- A duplicate COLLATING_SEQUENCE_NAME ({text}) was specified.

Condition Identifier: SME\$DUPLICATE_SEQN_NAME Code: SM 12

Description: The same name was specified for more than one collating sequence definition.

User Action: Rename one of the collating sequence definitions.

Further Information: See the SCL Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_NAME parameter. See the FORTRAN Usage manual for information on the SM5SEQN procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_NAME procedure.

--FATAL SM 13-- A COLLATING_SEQUENCE_NAME was not specified.

Condition Identifier: SME\$SEQN_REQUIRED Code: SM 13

Description: The collating sequence name must be specified at the beginning of the collating sequence specification.

User Action: For a command sort or merge, specify the COLLATING_SEQUENCE_NAME parameter. In a FORTRAN program, specify the SM5SEQN call; in a CYBIL program, specify the SMP\$COLLATING_NAME call.

Further Information: The COLLATING_SEQUENCE_NAME parameter is described in the NOS/VE Advanced File Management Usage manual. The SM5SEQN procedure is described in the FORTRAN Usage manual. The SMP\$COLLATING_SEQUENCE_NAME procedure is described in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL SM 14-- A duplicate COLLATING_SEQUENCE_ALTER parameter or procedure was specified.

Condition Identifier: SME\$DUPLICATE_SEQA Code: SM 14

Description: You cannot have more than one collating sequence alter step in a collating sequence definition.

User Action: Remove all except one collating sequence alter step.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_ALTER parameter. See the FORTRAN Usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SEQA procedures.

--FATAL SM 15-- The value ranges in the collating sequence definition list ({text}) are not all the same width.

Condition Identifier: SME\$SEQS_RANGE_DISCREPANCY Code: SM 15

Description: If the first value in the value step definition is a range, all subsequent values must be ranges and each range must have the same number of characters in it. The value step definition would then define the number of value steps equal to the width of the range (that is, if the range width is 3, it defines three value steps).

User Action: Correct the value step definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_STEP parameter.

--FATAL SM 16-- A range value in the collating sequence definition list ({text}) is incorrect.

Condition Identifier: SME\$INCORRECT_SEQS_RANGE Code: SM 16

Description: The lower bound of a range of characters must come before the upper bound of the range in the ASCII collating sequence.

User Action: Correct the range.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_STEP parameter.

--FATAL SM 17-- There cannot be more than 100 user-defined collating sequences.

Condition Identifier: SME\$TOO_MANY_SEQN Code: SM 17

Description: The maximum number of collating sequences you can define is 100 sequences per sort or merge. This includes collating sequences specified by load_collating_sequence and define_collating_sequence requests.

User Action: Remove the excess collating sequences.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_x and LOAD_COLLATING_TABLE parameters.

--FATAL SM 18-- Parameter value in collating sequence definition list ({text}) must be one character in length.

Condition Identifier: SME\$CHAR_REQUIRED Code: SM 18

Description: Each string can specify only one character (such as 'A').

User Action: Correct the string so that it contains only one character.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_STEP parameter. See the FORTRAN Language Definition Usage manual for information on the SM5SEQS procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_CHARACTERS procedure.

--FATAL SM 19-- ({text}) is not a SUM type name.

Condition Identifier: SME\$SUM_TYPE_REQUIRED Code: SM 19

Description: Each sum field definition must specify the data type of the field as a numeric format name.

User Action: Correct the data type name. The valid numeric data formats are: BINARY, BINARY_BITS, INTEGER, INTEGER_BITS, NUMERIC_FS, NUMERIC_LO, NUMERIC_LS, NUMERIC_NS, NUMERIC_TO, NUMERIC_TS, PACKED, and PACKED_NS.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM parameter. See the FORTRAN Usage manual for information on the SM5SUM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$SUM procedure.

--WARNING SM 20-- SUM overflow occurred on SUM field ({text}).

Condition Identifier: SME\$OVERFLOW_SUM_WARNING Code: SM 20

Description: The sum of the sum fields in two records with equal key values is too large to be stored in the sum field.

Unless the DE list option was specified, this message appears only once for this sum field, regardless of the number of records that cannot be summed.

User Action: None required. If an exception records file is specified, the records that cannot be summed are written to that file.

--FATAL SM 21-- ({file});duplicate DIRECTIVES_FILE name.

Condition Identifier: SME\$DUPLICATE_DIR_FILE Code: SM 21

Description: Each directive file can be specified only once for a sort or merge. The duplicate directive file specification could itself occur in a directive file.

User Action: Remove the duplicate directive file name.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on directive files.

--FATAL SM 22-- The number of KEY fields cannot be greater than 106.

Condition Identifier: SM\$TOO_MANY_KEY Code: SM 22

Description: Too many sort key fields are defined; the maximum number of key fields is 106.

User Action: Remove the excess key field definitions.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on key field definitions.

--FATAL SM 23-- The number of SUM fields cannot be greater than 100.

Condition Identifier: SM\$TOO_MANY_SUM Code: SM 23

Description: Too many sum fields are defined; the maximum number of sum fields is 100.

User Action: Remove excess sum fields.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5SUM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$SUM procedure.

--FATAL SM 24-- The collating sequence ({text}) has only one collating sequence definition.

Condition Identifier: SM\$SINGLE_CHAR_STEP Code: SM 24

Description: A collating sequence definition must specify more than one character.

User Action: Change the collating sequence definition so that it specifies at least two characters.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_STEP parameter. See the FORTRAN Usage manual for information on the SM5SEQS procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$COLLATING_CHARACTERS procedure.

--FATAL SM 25-- The number of DIRECTIVES_FILE entries cannot be greater than 100.

Condition Identifier: SM\$TOO_MANY_DIR_FILE Code: SM 25

Description: Too many directive files are specified; the maximum number of directive files is 100.

User Action: Reduce the number of directive files.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on directive files.

--FATAL SM 26-- A duplicate COLLATING_SEQUENCE_REMAINDER parameter or procedure was specified.

Condition Identifier: SM\$DUPLICATE_SEQR Code: SM 26

Description: You can only have one collating sequence remainder step in a collating sequence definition.

User Action: Remove all except one collating sequence remainder step.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_REMAINDER parameter. See the FORTRAN Usage manual for information on the SM5SEQR procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_REMAINDER call.

--FATAL SM 27-- Parameter ({text}) is not implemented.

Condition Identifier: SM\$NOT_PARAMETER Code: SM 27

Description: This parameter is not available under this release of Sort/Merge. {text} is one of the valid forms of the parameter, and not necessarily the parameter that was specified.

User Action: Remove the parameter.

Further Information: See the NOS/VE Advanced File Management Usage manual for a list of valid parameters.

--FATAL SM 28-- If LIST_OPTIONS is set to 'OFF', other LIST_OPTIONS values must not be given.

Condition Identifier: SME\$LIST_OP_ERROR Code: SM 28

Description: If the list option OFF or NONE is specified, no other list option can be specified for the sort or merge. For example, you cannot specify both the OFF and S list options for the same sort or merge.

User Action: Remove the OFF or NONE list option or remove all other list options.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the LIST_OPTIONS parameter. See the FORTRAN Language Definition Usage manual for information on the SM5LO procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$LIST_OPTION procedure.

--INFORMATIVE SM 29-- A LIST_OPTIONS value of DE is ignored when EXCEPTION_RECORDS_FILE processing has not been specified.

Condition Identifier: SME\$EXC_AND_DE Code: SM 29

Description: "Unused"formative message stating that the DE list option is ignored if you have not specified an exception records file.

User Action: None required.

Further Information: The LIST_OPTIONS and EXCEPTION_RECORDS_FILE parameters are described in the NOS/VE Advanced File Management Usage manual. The SM5LO and SM5ERF calls are described in the FORTRAN Usage manual. The SMP\$LIST_OPTION and SMP\$EXCEPTION_RECORDS_FILE calls are described in the CYBIL Keyed-File and Sort/Merge Interfaces manual.

--FATAL SM 30-- Incorrect range value for KEY ({text}) specified.

Condition Identifier: SME\$INCORRECT_KEY_RANGE Code: SM 30

Description: A range is specified when two periods (..) separate the first and second values in the key field definition. When a range is specified, the first value in the range (the lower bound) must be less than the second value in the range (the upper bound).

User Action: Correct the range.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY parameter format. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure.

--FATAL SM 31-- Record length from OWNCODE ({text}) is invalid.

Condition Identifier: SME\$WRONG_REC_LENGTH Code: SM 31

Description: The record length supplied by the owncode routine must be positive and must not conflict with the maximum record length specified for the sort.

User Action: Provide the correct record length. If you specified the OWNFL parameter or have called the SM5OFL or SMP\$OWNCODE_FIXED_RECORD_LENGTH procedure, the owncode routine record length must match the OWNFL, SM5OFL, or SMP\$OWNCODE_FIXED_RECORD_LENGTH value.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines and record length.

--FATAL SM 32-- Unable to transfer input record.

Condition Identifier: SME\$MOVE_ERROR Code: SM 32

Description: This is an operating system problem.

User Action: Follow your site procedures for reporting the software error.

--FATAL SM 33-- The RETURN_CODE supplied by OWNCODE ({text}) is not in the correct range.

Condition Identifier: SME\$BAD_RETURN_CODE Code: SM 33

Description: The return_code you supply must be in the range 0 through 3.

User Action: For owncode1 and owncode3 procedures, you can specify 0, 1, 2, or 3 as the return_code value. For owncode2, owncode4, and owncode5 procedures, the only valid return_codes values are 0 and 1.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines.

--FATAL SM 34-- SORT/MERGE unable to load OWNCODE ({text}). (Uppercase letters may be needed.)

Condition Identifier: SME\$OWNCODE_UNAVAIL Code: SM 34

Description: Sort/Merge could not find the module containing the owncode procedure. The reason for the failure could be:

- The object library file is not in the program object library list
- The entry point name was specified incorrectly. (The name must be specified using uppercase letters.)

User Action: Check the load map for the name of the procedure that could not be loaded. If necessary, use the DISPLAY_PROGRAM_ATTRIBUTES command to list the object library list, the SET_PROGRAM_ATTRIBUTES command to add an object library file to the list and/or the DISPLAY_OBJECT_LIBRARY command to display the contents of an object library.

Further Information: See the NOS/VE Object Code Management Usage manual for a description of the generation and specification of object library files.

--FATAL SM 35-- ({text}) is not defined as a user COLLATING_SEQUENCE_NAME.

Condition Identifier: SME\$UNDEFINED_USER_KEY_TYPE Code: SM 35

Description: The key type is not a predefined collating sequence or numeric format and has not been defined or loaded as a user-defined collating sequence.

User Action: Check the spelling of the collating sequence name. The predefined collating sequences are: ASCII, ASCII6, COBOL6, DISPLAY, EBCDIC, and EBCDIC6. The numeric data formats are: BINARY, BINARY_BITS, INTEGER, INTEGER_BITS, NUMERIC_FS, NUMERIC_LO, NUMERIC_LS, NUMERIC_NS, NUMERIC_TO, NUMERIC_TS, PACKED, PACKED_NS, and REAL.

If the collating sequence was to have been loaded, check that the name is the same as the key type name specified on the load request.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the collating sequence parameters. See the FORTRAN Language Definition Usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the collating sequence procedure calls.

--CATASTROPHIC SM 36-- SORT/MERGE internal call incompatibility in ({text}).

Condition Identifier: SME\$INTERNAL_INCOMPATIBILITY Code: SM 36

Description: This is an internal Sort/Merge error.

User Action: Follow your site procedures for reporting the software error.

--INFORMATIVE SM 37-- NODA is provided for Sort5 compatibility only and is ignored.

Condition Identifier: SME\$NODA_IGNORED Code: SM 37

Description: Informative message stating that NOS/VE Sort/Merge recognizes the SM5NODA call, but does not use the specification.

User Action: None.

--CATASTROPHIC SM 38-- Memory not available.

Condition Identifier: SME\$NO_MEMORY Code: SM 38

Description: This is an operating system problem. The system could not allocate sufficient memory.

User Action: Follow your site procedures for reporting the software error.

--FATAL SM 39-- Procedure call out of sequence.

Condition Identifier: SME\$BAD_CALL_SEQUENCE Code: SM 39

Description: One of the Sort/Merge procedures is out of order.

User Action: Correct the procedure call order. The SM5SORT, SMPBEGIN_SORT_SPECIFICATION, SM5MERG, or SMP\$BEGIN_MERGE_SPECIFICATION procedure must be called first. The SM5END or SMP\$END_SPECIFICATION procedure must be called last.

Further Information: See the appropriate language usage manual or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on procedure calls.

--FATAL SM 40-- An invalid number of FROM files was specified.

Condition Identifier: SME\$IN_FILE_NUMBER Code: SM 40

Description: The number of input files cannot exceed 100.

User Action: Reduce the number of input files to 100. Request a second sort or merge for the other files and then merge the two output files.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on input files using command calls. See the FORTRAN Language Definition Usage manual for information on input files using procedure calls. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on input files using procedure calls.

--FATAL SM 41-- ({text}) is not a COLLATING_SEQUENCE_REMAINDER or COLLATING_SEQUENCE_ALTER value.

Condition Identifier: SME\$SEQR_SEQA_VALUE Code: SM 41

Description: The specified value is not recognized as a boolean value.

User Action: Specify YES or NO on a COLLATING_SEQUENCE_REMAINDER or COLLATING_SEQUENCE_ALTER parameter. Specify 'YES' or 'NO' on a SM5SEQR, SM5SEQA, SMP\$COLLATING_REMAINDER, or SMP\$COLLATING_ALTER procedure call.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_REMAINDER and COLLATING_SEQUENCE_ALTER parameters. See the FORTRAN Usage manual for information on the SM5SEQR and SM5SEQA procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$COLLATING_REMAINDER and SMP\$COLLATING_ALTER procedures.

--FATAL SM 42-- Invalid name for SM5SEQN or SMP\$COLLATING_NAME.

Condition Identifier: SME\$INVALID_SEQN Code: SM 42

Description: The name is not a valid collating sequence name.

A valid SCL name has from 1 through 31 characters and begins with a nondigit. The valid characters are all letters, digits, and the special characters # @ _ \$ [] { } \ | ~ ^.

The name cannot be the name of a collating sequence already defined for the sort or merge. The predefined collating sequence names are: ASCII, ASCII6, COBOL6, DISPLAY, EBCDIC, and EBCDIC6.

User Action: Determine the names of the collating sequences already defined and then specify a new and valid name for the collating sequence.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the COLLATING_SEQUENCE_NAME parameter. See the FORTRAN Usage manual for information on the SM5SEQN procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$COLLATING_NAME procedure.

--FATAL SM 43-- Characters are duplicated in collating sequence ({text}).

Condition Identifier: SM\$DUP_SEQS_CHARACTER Code: SM 43

Description: The collating sequence definition specifies the same character more than once. The duplication could result if the character is inadvertently included in a range. A range specified all characters in the ASCII collating sequence included in the range bounds.

User Action: Remove the duplicate character specification.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on defining Sort/Merge collating sequences. See the FORTRAN Language Definition Usage manual for information on the SM5SEQS procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$COLLATING_CHARACTERS procedure.

--WARNING SM 44-- Unable to write a record to the TO file. This record has been written to the EXCEPTION_RECORDS_FILE.

Condition Identifier: SM\$PUT_NEXT_ERROR Code: SM 44

Description: When an exception records file has been specified, Sort/Merge writes each record that it cannot write to the output (TO) file to the exception records file instead.

User Action: You should list the exception records file to see the records that were removed from the sort or merge.

Further Information: See the NOS/VE Advanced File Management, FORTRAN Usage, or CYBIL Keyed-File & Sort/Merge manual for more information on invalid records.

--FATAL SM 45-- Unable to write a record to the TO file.

Condition Identifier: SM\$TO_FILE_ERROR Code: SM 45

Description: Sort/Merge could not write a record to the output (TO) file. Because no exception records file was specified, the sort or merge terminates due to the error.

User Action: Check the data in the input files to try to determine the cause of the write errors. Re-run the sort or merge, specifying an exception records file so that the invalid records can be listed.

Further Information: For the EXCEPTION_RECORDS_FILE parameter description, see the NOS/VE Advanced File Management Usage manual. For the SM5ERF call description, see the FORTRAN Usage manual. For the SMP\$EXCEPTION_RECORDS_FILE call description, see the CYBIL Keyed-File and Sort/Merge manual.

--FATAL SM 46-- Unable to write a record to the TO file and to the EXCEPTION_RECORDS_FILE.

Condition Identifier: SM\$TO_AND_EXC_ERROR Code: SM 46

Description: Sort/Merge could not write a record to either the output (TO) file or the exception records file.

User Action: Check the other Sort/Merge messages and the input files to determine the cause of the write error. If appropriate, follow your site procedures for reporting the problem.

--WARNING SM 47-- Invalid KEY field data in KEY field ({text}).

Condition Identifier: SM\$KEY_UNORDERED Code: SM 47

Description: The data in the key field does not correspond with the description given to Sort/Merge. If the key type is a binary numeric format, the data must be numeric. If the key type is a numeric character format, the data must be numeric characters. If the key type is a collating sequence name, the data must be represented by ASCII character codes.

User Action: Correct the key field definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on valid key field data.

--WARNING SM 48-- Invalid SUM field data in SUM field ({text}).

Condition Identifier: SME\$INVALID_SUM_FIELD Code: SM 48

Description: Invalid sum field data was found in sum field. Data found does not correspond with the description given to Sort/Merge.

Unless the DE list option was specified, this message appears only once for this sum field, regardless of the number of records containing invalid data in the sum field.

User Action: Check the records written to the exception records file if such a file was specified. As appropriate, change the data type in the sum field definition or correct the invalid data in the records.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5SUM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on SMP\$SUM procedure.

--WARNING SM 49-- Incorrect parameter for SM5LO or SMP\$LIST_OPTION is ignored.

Condition Identifier: SME\$UNRECOGNIZED_LIST_OP Code: SM 49

Description: The incorrect parameter for SM5LO or SMP\$LIST_OPTION is ignored.

User Action: Check that correct parameter was specified.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the LIST_OPTIONS parameter. See the FORTRAN Language Definition Usage manual for information on the SM5LO procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$LIST_OPTIONS procedure.

--FATAL SM 50-- RETAIN_ORIGINAL_ORDER and SUM are mutually exclusive.

Condition Identifier: SME\$SUM_AND_RETAIN_ILLEGAL Code: SM 50

Description: A sort or merge can specify only one method of processing records with equal key values.

Currently, it specifies both that the records should be summed and that their original order should be retained.

User Action: Remove either the retain_original_order or summing specification.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the RETAIN_ORIGINAL_ORDER and SUM parameters. See the FORTRAN Usage manual for information on the SM5RETA and SM5SUM procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$RETAIN_ORIGINAL_ORDER and SMP\$SUM procedures.

--WARNING SM 51-- Parameter for SM5EL or SMP\$error_level not recognized.

Condition Identifier: SME\$SM5EL_PARAMETER_ERROR Code: SM 51

Description: The error level value was not recognized; the default minimum error level, informational, is assumed.

User Action: Specify a valid error level value. The valid values are T for trivial, I for informational, W for warning, F for error and fatal, C for catastrophic, and NONE for none.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the ERROR_LEVEL parameter. See the FORTRAN Language Definition Usage manual for information on the SM5EL procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$error_level procedure.

--FATAL SM 52-- OWNCODE ({text}) cannot be used in a MERGE.

Condition Identifier: SME\$OWNCODE1_2_IN_MERGE Code: SM 52

Description: The owncode 1 and owncode 2 procedures can only be used for sorting.

User Action: Remove the owncode 1 or owncode 2 procedure specifications.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines.

--WARNING SM 53-- FILE_CONTENTS for TO file ignored.

Condition Identifier: SME\$FILE_CONTENTS_INVALID Code: SM 53

Description: The FILE_CONTENTS attribute value for the output file has been ignored.

User Action: None required. To display the FILE_CONTENTS attribute value, enter this command, specifying the output file:

DISPLAY_FILE_ATTRIBUTES, file, DISPLAY_OPTION=FILE_CONTENTS

Further Information: See the NOS/VE System Usage manual for the DISPLAY_FILE_ATTRIBUTES command description and the list of valid file_contents values (in the SET_FILE_ATTRIBUTES command description).

--WARNING SM 54-- An indefinite REAL number was found in KEY field ({text}).

Condition Identifier: SME\$INDEFINITE_NUMBER Code: SM 54

Description: An indefinite real number cannot be ordered in an output file.

This message is issued only once for this key field, even if more than one indefinite real number was found.

User Action: Check that the key field definitions are correct.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY parameter. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure.

--FATAL SM 55-- Record length from OWNCODE ({text}) is different from the OWNCODE_FIXED_LENGTH value.

Condition Identifier: SME\$NOT_SAME_AS_OFI Code: SM 55

Description: If a fixed record length is specified on the OWNCODE_FIXED_LENGTH parameter or an SM5OFI or SMP\$OWNCODE_FIXED_RECORD_LENGTH procedure call, the length of all records from all owncode procedures must be that value.

User Action: Correct the fixed record length specification or correct the owncode procedure.

Further Information: See the NOS/VE Advanced File Management Usage manual, the FORTRAN Language Definition Usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines.

--WARNING SM 56-- File ({file}) is opened at EOI.

Condition Identifier: SME\$INPUT_FILE_OPEN_AT_EOI Code: SM 56

Description: The input file specified is opened at EOI; the file will not be processed.

User Action: Check that the file contains data; then correct the open position specification for the file.

Further Information: See the NOS/VE System Usage manual for a description of the open_position attribute. See the SCL Advanced File Management Usage manual for information on the FROM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5FROM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$FROM_FILES procedure.

--INFORMATIVE SM 57-- Sort/Merge found ({text}) records that were too short to contain one or more of the SUM and KEY fields.

Condition Identifier: SME\$SHORT_RECORD Code: SM 57

Description: Informative message giving the number of short records.

If an exception records file is specified, the short records are written to it and removed from the sort or merge.

User Action: None required.

Further Information: See the NOS/VE Advanced File Management Usage, FORTRAN Usage, or CYBIL Keyed-File and Sort/Merge Interfaces manual for information on sum and key fields.

--INFORMATIVE SM 58-- The number of zero-length records appended to the end of the TO file is ({text}).

Condition Identifier: SME\$ZERO_LENGTH Code: SM 58

Description: Zero-length records are records with no length and containing no data. Zero-length records are output last on the output file. Records containing all blanks could become zero-length records if blank compression is performed.

User Action: None required.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces for information on record length.

--FATAL SM 59-- The KEY or SUM position ({text}) is not a positive integer.

Condition Identifier: SME\$WRONG_POSITION Code: SM 59

Description: The value specified as the position of a sum or key field in a record must be a positive integer.

User Action: Correct the position value.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY and SUM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5KEY and SM5SUM procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY and SMP\$SUM procedures.

--FATAL SM 60-- Length of SUM or KEY field must be positive.

Condition Identifier: SME\$SHORT_KEY_LENGTH Code: SM 60

Description: The value specified as the length of a sum or key field must be positive.

User Action: Correct the length value.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM and KEY parameters. See the FORTRAN Language Definition Usage manual for information on the SM5SUM or SM5KEY procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces for information on the SMP\$KEY and SMP\$SUM procedures.

--FATAL SM 61-- KEY field ({text}) length exceeds limit of {text} bytes.

Condition Identifier: SME\$LONG_KEY_LENGTH Code: SM 61

Description: The maximum key length is 1023 bytes (8184 bits).

User Action: Correct the key length.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on key fields. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure.

--FATAL SM 62-- The parameter for the C170_COMPATIBLE option is not a recognizable parameter.

Condition Identifier: SME\$WRONG_C170 Code: SM 62

Description: The value for the C170_COMPATIBLE specification is not recognized as a boolean value.

User Action: Correct the value. For the SCL parameter, the valid boolean values are YES and NO. For a procedure call, the valid boolean values are YES and NO.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the C170_COMPATIBLE parameter. See the FORTRAN Usage manual for information on the SM5CC procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$C170_COMPATIBLE procedure.

--FATAL SM 63-- The length of a REAL KEY must be 8 or 16 bytes; ({text}) found.

Condition Identifier: SME\$REAL_KEY_LENGTH Code: SM 63

Description: A key field defined with key type REAL must be one word (8 bytes) long.

User Action: Change the key field length to 8.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on key field lengths.

--INFORMATIVE SM 64-- KEY fields overlap other KEY fields.

Condition Identifier: SME\$KEY_OVERLAP_KEY_FIELD Code: SM 64

Description: Your key field definitions are not correct. You can specify up to 106 key fields. Overlapping key fields are invalid when they overlap sum fields or key fields that are ordered by collating sequences defined with the alter option.

User Action: Check the syntax of your key field specification. Ensure that range, hierarchy, and length formats are correct.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the KEY parameter.

--FATAL SM 65-- KEY fields cannot overlap SUM fields.

Condition Identifier: SME\$NO_KEY_SUM_OVERLAP Code: SM 65

Description: KEY fields and SUM fields cannot overlap one another. That is, the fields described as KEY fields cannot also be SUM fields.

User Action: Correct the overlapping fields.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM and KEY parameters. See the FORTRAN Usage manual for information on the SM5SUM and SM5KEY procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$SUM and SMP\$KEY procedures.

--FATAL SM 66-- SUM fields cannot overlap other SUM fields.

Condition Identifier: SME\$NO_SUM_FIELDS_OVERLAP Code: SM 66

Description: SUM fields cannot overlap one another.

User Action: Correct the overlapping fields.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM parameter. See the FORTRAN Usage manual for information on the SM5SUM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$SUM procedure.

--FATAL SM 67-- Collating sequences which use the alter option cannot be used in overlapping KEY fields.

Condition Identifier: SME\$NO_ALTER_WITH_KEY_OVERLAP Code: SM 67

Description: You cannot overlap KEY fields that have been ordered by a collating sequence defined with the collating sequence alter step.

User Action: Correct the overlapping key fields.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY and COLLATING_SEQUENCE_ALTER parameters. See the FORTRAN Usage manual for information on the SM5KEY and SM5SEQA procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY and SMP\$COLLATING_ALTER procedures.

--FATAL SM 68-- OWNCODE 5 and SUM are mutually exclusive.

Condition Identifier: SME\$OWN5_AND_SUM_ILLEGAL Code: SM 68

Description: An owncode 5 routine and sum fields cannot be specified for the same sort or merge. You can specify only one way of processing records with matching key values.

User Action: Either remove the owncode 5 routine specification or the sum field definitions.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on summing and owncode routines. See the FORTRAN Language Definition Usage manual for information on the SM5SUM procedure and owncode routines. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$SUM procedure and owncode routines.

--FATAL SM 69-- The KEY length cannot be determined.

Condition Identifier: SME\$NO_KEY_LENGTH Code: SM 69

Description: The default value for the default key field is the smallest minimum record length of any input file. If the minimum record length attribute has not been specified for an input file, its value is zero. Sort/Merge cannot define a key field with length zero.

User Action: Either explicitly define a key field or specify a nonzero value for the minimum record length attribute of all input files.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY parameter and for information on specifying record length. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure and setting file attributes. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure and setting file attributes.

--FATAL SM 70-- Either OWNCODE_MAXIMUM_RECORD_LENGTH or OWNCODE_FIXED_LENGTH must be specified.

Condition Identifier: SME\$NO_MRL_INFO Code: SM 70

Description: If there are no input and output files, either owncode maximum record length or owncode fixed length must be specified to set the maximum record length.

User Action: Provide the owncode maximum or owncode fixed length.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the OWNCODE_MAXIMUM_LENGTH OR OWNCODE_FIXED_LENGTH parameters. See the FORTRAN Language Definition manual for information on the SM5OFL or SM5OMRL procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$OWNCODE_FIXED_RECORD_LENGTH or SMP\$OWNCODE_MAX_RECORD_LENGTH procedures.

--FATAL SM 71-- KEY field ({text}) beyond maximum record length.

Condition Identifier: SME\$KEY_OUT_OF_RANGE Code: SM 71

Description: The length of a key field is greater than the maximum record length.

User Action: Correct the key field definition or the maximum record length.

To see the maximum_record_length of the file, enter the file reference in this command:

DISPLAY_FILE_ATTRIBUTES, file, DISPLAY_OPTION=MAXIMUM_RECORD_LENGTH

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the KEY parameter and for information on record length. See the FORTRAN Language Definition Usage manual for information on the SM5KEY procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$KEY procedure. See the SCL System Interface Usage manual for a description of the maximum_record_length attribute.

--FATAL SM 72-- OWNCODE_MAXIMUM_RECORD_LENGTH and OWNCODE_FIXED_LENGTH cannot be specified together.

Condition Identifier: SME\$OWNFL_AND_OWNMRL_ILLEGAL Code: SM 72

Description: You cannot specify both an owncode fixed record length and an owncode maximum record length for the same sort or merge.

User Action: Remove one of the owncode record length specifications.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the OWNCODE_FIXED_LENGTH and OWNCODE_MAXIMUM_LENGTH parameters. See the FORTRAN Language Definition Usage manual for information on the SM5OFL and SM5OMRL procedures. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$OWNCODE_FIXED_RECORD_LENGTH or SMP\$OWNCODE_MAX_RECORD_LENGTH procedure.

--WARNING SM 73-- VERIFY_MERGE_INPUT_ORDER is ignored in a SORT.

Condition Identifier: SME\$VERIFY_IN_SORT Code: SM 73

Description: Verification that the merge input records are sorted is valid only for a merge; the verify parameter or procedure is ignored for a sort.

User Action: None required, although you should remove the verify parameter or procedure from the sort specification.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the VERIFY_MERGE_INPUT_ORDER parameter.

--WARNING SM 74-- MERGE FROM file ({file}) out of order.

Condition Identifier: SM\$MERGE_INPUT_OUT_OF_ORDER Code: SM 74

Description: The order verification has detected a record that is out of order in a merge input file. Files to be merged must be in sorted order.

Unless the DE list option was specified, this message appears only once for this file, regardless of the number of records that are out of order.

User Action: Check the records written to the exception records file if an exception records file was specified. Sort the input files and then reattempt the merge.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the EXCEPTION parameter. See the FORTRAN Language Definition Usage manual for information on the SM5EXC procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manuals for information on the SMP\$EXCEPTION_FILE procedure.

--FATAL SM 75-- The parameter for the RETAIN_ORIGINAL_ORDER option ({text}) is not a recognizable parameter.

Condition Identifier: SM\$WRONG_RETAIN Code: SM 75

Description: The value for the retain_original_order specification is not recognized as a boolean value.

User Action: Correct the value. For the SCL parameter, the valid boolean values are YES and NO. For a procedure call, the valid boolean values are 'YES' and 'NO'.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the RETAIN_ORIGINAL_ORDER parameter. See the FORTRAN Usage manual for information on the SM5RETA procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$RETAIN_ORIGINAL_ORDER procedure.

--FATAL SM 76-- The parameter for the VERIFY_MERGE_INPUT_ORDER option ({text}) is not a recognizable parameter.

Condition Identifier: SM\$WRONG_VERIFY Code: SM 76

Description: The value on the verify specification is not recognized as a boolean value.

User Action: Correct the value. For the SCL parameter, the valid boolean values are YES and NO. For a procedure call, the valid boolean values are 'YES' and 'NO'.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the VERIFY parameter. See the FORTRAN Language Definition Usage manual for information on the SM5VER procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$VERIFY procedure.

--WARNING SM 77-- FROM file ({file}) was not found.

Condition Identifier: SM\$FROM_FILE_NOT_FOUND Code: SM 77

Description: A file of records to be sorted or merged was not found. The sort or merge continues without that input file.

User Action: Check that the file was specified correctly and that the file is accessible.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the FROM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5FROM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$FROM_FILES procedure.

--WARNING SM 78-- DIRECTIVES_FILE ({file}) was not found.

Condition Identifier: SM\$DIRECTIVE_FILE_NOT_FOUND Code: SM 78

Description: The directive file specified could not be found. The sort or merge is performed without the directives from the file.

User Action: Check that the correct file name was specified and that the file is accessible. If appropriate, enter a DISPLAY_CATALOG_ENTRY command to display the permits for the directives file.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on directive files. See the SCL System Interface Usage manual for a description of the DISPLAY_CATALOG_ENTRY command.

--CATASTROPHIC SM 79-- ALLOCATE in system heap fails for ({text}).

Condition Identifier: SME\$SYSTEM_HEAP_OUT_OF_SPACE Code: SM 79

Description: Internal error; the system was not able to allocate space for the variable.

User Action: Reattempt the sort or merge when the system is less busy.

--CATASTROPHIC SM 80-- PUSH onto system stack fails for ({text}).

Condition Identifier: SME\$STACK_OUT_OF_SPACE Code: SM 80

Description: Internal error; the system was not able to allocate space for the variable.

User Action: Reattempt the sort or merge when the system is less busy.

--CATASTROPHIC SM 81-- SORT internal error: NEXT sequence allocation fails for ({text}).

Condition Identifier: SME\$SEQUENCE_OUT_OF_SPACE Code: SM 81

Description: Internal error; Sort/Merge was not able to allocate space for the variable.

User Action: Follow site procedures to report the software error.

--FATAL SM 82-- SUM field ({text}) is too large.

Condition Identifier: SME\$SUM_FIELD_TOO_LARGE Code: SM 82

Description: The maximum size of the binary sum fields is one word. The maximum size of the character (NUMERIC_LO, NUMERIC_LS, NUMERIC_TO, NUMERIC_TS, NUMERIC_NS, or NUMERIC_FS) sum fields with a nonseparate sign is 17 digits. If the sum fields have a separate sign, the maximum size is 17 digits plus one digit for the sign.

User Action: Correct the length in the sum field definition.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the SUM parameter. See the FORTRAN Language Definition Usage manual for information on the SM5SUM procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information for the SMP\$SUM procedure.

--INFORMATIVE SM 83-- The nonembedded key in TO file ({file}) is removed from the start of the record.

Condition Identifier: SME\$NONEMBED_TO_KEY Code: SM 83

Description: This is an informative message. This condition will not affect your sort or merge.

User Action: No action is required.

--FATAL SM 84-- The major SORT/MERGE KEY position differs from the PRIMARY_KEY position of the indexed sequential TO file.

Condition Identifier: SME\$BAD_PRIMARY_SORT_KEY Code: SM 84

Description: When the FILE_ORGANIZATION attribute of the output (TO) file is INDEXED_SEQUENTIAL, the position specified in the first key field definition must be the same position specified by the KEY_POSITION attribute value for the file. This is because the records must be sorted first by the primary-key value before any other sort keys.

User Action: Change the first value in the first key field definition to the KEY_POSITION attribute value plus one. (This is because the KEY_POSITION value counts byte positions beginning with 0 while Sort/Merge counts byte positions beginning with 1.)

To see the KEY_POSITION attribute for the file, enter the file reference in this command:

```
DISPLAY_FILE_ATTRIBUTE, file, DISPLAY_OPTION=KEY_POSITION
```

Further Information: See the KEY and KEY_POSITION parameter descriptions in the SCL Advanced File Management manual. See the DISPLAY_FILE_ATTRIBUTE command description in the SCL Quick Reference manual.

--WARNING SM 85-- The length of the major SORT/MERGE KEY was changed to the length of the PRIMARY_KEY ({text} bytes) of the TO file.

Condition Identifier: SM\$BAD_PRIMARY_SORT_KEY_LENGTH Code: SM 85

Description: Because the FILE_ORGANIZATION attribute of the output (TO) file is INDEXED_SEQUENTIAL, the key length specified in the first key field definition must match the KEY_LENGTH attribute value for the file.

User Action: None required, although you should change the sort key length specification to the value given in the message or change the KEY_LENGTH attribute value for the output file.

--WARNING SM 86-- The KEY type of the major SORT/MERGE KEY was changed to the KEY type ({text} type) of the PRIMARY_KEY of the TO file.

Condition Identifier: SM\$BAD_PRIMARY_SORT_KEY_TYPE Code: SM 86

Description: Because the FILE_ORGANIZATION attribute of the output (TO) file is INDEXED_SEQUENTIAL, the data type specified in the first key field definition must correspond to the KEY_TYPE attribute of the file. If the KEY_TYPE is integer, the sort key type must be INTEGER. Otherwise, the sort key type must be a collating sequence name or a numeric data format using a character representation.

User Action: None required, although you should change the key type in the key field definition or the KEY_TYPE attribute value for the output file.

--FATAL SM 87-- The major SORT/MERGE KEY must be sorted in ascending order for an indexed sequential TO file.

Condition Identifier: SM\$DESCENDING_PRIMARY_SORT_KEY Code: SM 87

Description: When the FILE_ORGANIZATION attribute of the output (TO) file is INDEXED_SEQUENTIAL, the sort order specified in the first key field definition must be ascending (A). This is because the first key field definition must specify the primary key of the file and the records in an indexed-sequential file are always kept in ascending primary-key value order.

User Action: Change the sort order specified in the first key field definition to A (ascending).

Further Information: See the NOS/VE Advanced File Management Usage, the appropriate language usage manual, or CYBIL Keyed-File and Sort/Merge manual.

--FATAL SM 88-- Direct access files are not permitted as the TO file. Use CREATE_KEYED_FILE to convert the TO file to a direct access file.

Condition Identifier: SM\$TO_DIRECT_ACCESS Code: SM 88

Description: The output (TO) file cannot be a direct-access file. If appropriate, use the COPY_KEYED_FILE command to convert the TO file to a direct-access file after the Sort/Merge is complete.

User Action: Change the output (TO) file to a file organization other than direct access.

Further Information: See the NOS/VE Advanced File Management Usage manual for information on the TO parameter. See the FORTRAN Usage manual for information on the SM5TO procedure. See the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the SMP\$TO_FILE procedure.

--FATAL SM 89-- The TO file cannot be the same as a FROM file in a MERGE.

Condition Identifier: SM\$TO_FROM Code: SM 89

Description: Sort/Merge input (FROM) files can be direct_access files. The output (TO) file cannot be a direct_access file.

User Action: Sort/Merge provides several options for the output file. If you omit the TO parameter and the OWNCODE_PROCEDURE_3 parameter, Sort/Merge writes all output records to file NEW.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on the TO and FROM parameters.

--WARNING SM 90-- The value for SMV\$MEMORY_USAGE_LIMIT ({text}) was less than 64 or greater than 16383 and was treated as 64 or 16383 respectively.

Condition Identifier: SME\$BAD_MUL Code: SM 90

Description: You specified an SMV\$MEMORY_USAGE_LIMIT outside the range of 64 through 16,383.

User Action: No action is necessary. If you specify an integer less than 64, Sort/Merge uses 64. If you specify an integer greater than 16,383, Sort/Merge uses 16,383.

--FATAL SM 92-- OMIT_DUPLICATES and RETAIN_ORIGINAL_ORDER are mutually exclusive.

Condition Identifier: SME\$OMIT_AND_RETAIN_ILLEGAL Code: SM 92

Description: Each sort or merge can specify only one method of processing records with equivalent key values.

User Action: Select OMIT_DUPLICATES, OWNCODE_PROCEDURE_5, RETAIN_ORIGINAL_ORDER, or SUM. These parameters are mutually exclusive.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on processing records with equivalent key values.

--FATAL SM 93-- OMIT_DUPLICATES and SUM are mutually exclusive.

Condition Identifier: SME\$OMIT_AND_SUM_ILLEGAL Code: SM 93

Description: Each sort or merge can specify only one method of processing records with equivalent key values.

User Action: Select OMIT_DUPLICATES, OWNCODE_PROCEDURE_5, RETAIN_ORIGINAL_ORDER, or SUM. These parameters are mutually exclusive.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on processing records with equivalent key values.

--FATAL SM 94-- OMIT_DUPLICATES and OWNCODE 5 are mutually exclusive.

Condition Identifier: SME\$OWN5_AND_OMIT_ILLEGAL Code: SM 94

Description: Each sort or merge can specify only one method of processing records with equivalent key values.

User Action: Select OMIT_DUPLICATES, OWNCODE_PROCEDURE_5, RETAIN_ORIGINAL_ORDER, or SUM. These parameters are mutually exclusive.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on processing records with equivalent key values.

--FATAL SM 95-- The parameter ({text}) for the OMIT_DUPLICATES option is not recognizable.

Condition Identifier: SME\$WRONG_OMIT Code: SM 95

Description: Each sort or merge can specify only one method of processing records with equivalent key values. Therefore, the OMIT_DUPLICATES, OWNCODE_PROCEDURES_5, RETAIN_ORIGINAL_ORDER, and SUM parameters are mutually exclusive.

User Action: If you select the OMIT_DUPLICATES parameter, specify OMIT_DUPLICATES=boolean.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on equivalent key values.

--FATAL SM 96-- The parameter value ({text}) for the ZERO_LENGTH_RECORDS option is not recognizable.

Condition Identifier: SME\$BAD_ZLR_PARM Code: SM 96

Description: The ZERO_LENGTH_RECORDS parameter provides the following keywords: DELETE, PAD, and LAST.

User Action: Check the parameter value for the ZERO_LENGTH_RECORDS option.

--INFORMATIVE SM 97-- The nonembedded key in FROM file ({file}) is inserted at the start of the record.

Condition Identifier: SME\$NONEMBED_FROM_KEY Code: SM 97

Description: This is an informative message. This condition will not affect your sort or merge.

User Action: No action is required.

--FATAL SM 98-- The RECORD_TYPE parameter ({text}) for the FROM memory area option is not recognizable.

Condition Identifier: SME\$BAD_FMA_PARM Code: SM 98

Description: The FROM parameter on the Sort/Merge command permits the following RECORD_TYPE parameters: FIXED (F) or VARIABLE (V).

User Action: Ensure that you have selected a valid RECORD_TYPE parameter. UNDEFINED (U) or TRAILING (T) are invalid RECORD_TYPES.

Further Information: See the SCL Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on RECORD_TYPES.

--FATAL SM 99-- The RECORD_TYPE parameter ({text}) for the TO memory area option is not recognizable.

Condition Identifier: SME\$BAD_TMA_PARM Code: SM 99

Description: The TO parameter on the Sort/Merge command permits the following RECORD_TYPE parameters: FIXED (F) or VARIABLE (V).

User Action: Ensure that you have selected a valid RECORD_TYPE parameter. UNDEFINED (U) or TRAILING (T) are invalid RECORD_TYPES.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on RECORD_TYPES.

--FATAL SM 100-- The TO memory area option and the TO option are mutually exclusive.

Condition Identifier: SME\$TMA_TO_FILE Code: SM 100

Description: The TO option specifies the file to which sorted or merged records are written if records are left after owncode processing. The TO_MEMORY_AREA option specifies a memory area to be used as the destination for output records.

User Action: A sort/Merge specification can specify only one destination for output records. The destination can be a file or a memory area, but not both. Specify only one option for your sort or merge.

--FATAL SM 101-- OWNCODE1 and OWNCODE2 procedures cannot be used with a FROM memory area.

Condition Identifier: SME\$FMA_OWNCODE Code: SM 101

Description: If you specify an OWNCODE1 or OWNCODE2 routine, the return_code, reca, and r/a parameters are passed by Sort/Merge. The FROM parameter is not used.

User Action: Ensure that your input data conforms to the OWNCODE_PROCEDURE_n=entry_point_name format.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines.

--FATAL SM 102-- OWNCODE3 and OWNCODE4 procedures cannot be used with a TO memory area.

Condition Identifier: SME\$TMA_OWNCODE Code: SM 102

Description: If you specify an OWNCODE3 or OWNCODE4 routine, the return_code, reca, and r/a parameters are passed by Sort/Merge. The TO parameter is not used.

User Action: Ensure that your input data conforms to the OWNCODE_PROCEDURE_n=entry_point_name format.

Further Information: See the NOS/VE Advanced File Management Usage manual, the appropriate language usage manual, or the CYBIL Keyed-File and Sort/Merge Interfaces manual for information on owncode routines.

--FATAL SM 103-- The length of a REAL SUM field must be 8 or 16 bytes; {{text}} found.

Condition Identifier: SME\$BAD_REAL_SUM Code: SM 103

Description: Numeric data formats for the REAL SUM field must be normalized floating-point numbers, either single-precision (8 bytes) or double-precision (16 bytes).

User Action: Check the numeric data formats of the REAL SUM field.

--WARNING SM 104-- An indefinite REAL number was found in SUM field {{text}}.

Condition Identifier: SME\$INDEF_REAL_SUM Code: SM 104

Description: Numeric data formats for the REAL SUM field must be normalized floating-point numbers, either single-precision (8 bytes) or double-precision (16 bytes).

User Action: Check the numeric data formats of the REAL SUM field.

--WARNING SM 105-- The number of records with partial numeric KEY fields is {{text}}.

Condition Identifier: SME\$PARTIAL_KEY Code: SM 105

Description: Sort/Merge processing differs when the field is only partially beyond the end of the record. If the partial field is a numeric key field, Sort/Merge processes it as an exception.

User Action: Check your exception records, reenter those records, and reexecute your Sort/Merge process.

--WARNING SM 106-- The number of records with partial SUM fields is {{text}}.

Condition Identifier: SME\$PARTIAL_SUM Code: SM 106

Description: Sort/Merge processing differs when the field is only partially beyond the end of the record. If the partial field is a sum field, Sort/Merge processes it as an exception.

User Action: Check your exception records, reenter those records, and reexecute your Sort/Merge process.

--WARNING SM 107-- Merge input segment out of order at PVA {{text}}.

Condition Identifier: SME\$MERGE_SEG_OUT_OF_ORDER Code: SM 107

Description: Sort/Merge encountered a record out of order. If you specified an exception records file, the out-of-order record has been written to the exception records file and deleted from the merge operation.

User Action: Check your input records and your command format, then repeat the Sort/Merge process.

--WARNING SM 108-- FROM file {{file}} was empty and not previously opened.

Condition Identifier: SME\$NEW_FILE_EMPTY Code: SM 108

Description: The specified file has been created by the CREATE_FILE command, but has not been opened and does not contain data.

User Action: Store data into the file by using the COLLECT_TEXT command, the EDIT_FILE command, or a program.

--ERROR SW 6-- The catalog specified by the ENVIRONMENT_CATALOG parameter contains a subcatalog named \$DESCRIPTION which is expected to be a file.

Condition Identifier: SWE\$DESCRIPTION_IS_A_CATALOG Code: SW 6

Description: A PPE environment catalog must contain a file named \$DESCRIPTION. The \$DESCRIPTION file contains information about the state of your environment catalog from the last time you exited it. However, your environment catalog contains a catalog named \$DESCRIPTION, which causes PPE to fail when you specify your environment catalog on the ENTER_PPE command.

User Action: Change the name of the catalog named \$DESCRIPTION in your environment catalog then enter PPE, specifying your environment catalog on the ENVIRONMENT_CATALOG parameter. PPE will ask you whether you want to create the \$DESCRIPTION file with default parameters. Indicate yes to continue the PPE session.

--ERROR SW 7-- The \$LOCAL catalog must not be the value of the ENVIRONMENT_CATALOG parameter.

Condition Identifier: SWE\$SEC_CANT_BE_\$LOCAL Code: SW 7

Description: You specified the \$LOCAL catalog on the ENVIRONMENT_CATALOG parameter of the ENTER_PPE command. The environment catalog must be a catalog other than \$LOCAL or \$USER.

User Action: Specify another catalog name on the ENVIRONMENT_CATALOG parameter.

--ERROR SW 8-- The ENVIRONMENT_CATALOG parameter must specify a catalog, but {text} is a file.

Condition Identifier: SWE\$SEC_MUST_BE_CATALOG Code: SW 8

Description: You specified a file name on the ENVIRONMENT_DESCRIPTION parameter of the ENTER_PPE command but the ENVIRONMENT_DESCRIPTION parameter must be a catalog.

User Action: Specify a catalog name on the ENVIRONMENT_DESCRIPTION parameter.

--ERROR SW 21-- The catalog specified by the ENVIRONMENT_CATALOG parameter is expected to contain a file named \$DESCRIPTION, but it does not.

Condition Identifier: SWE\$MISSING_DESCRIPTION Code: SW 21

Description: The environment catalog you specified on the ENVIRONMENT_DESCRIPTION parameter of the ENTER_PPE command does not contain a file named \$DESCRIPTION. The environment catalog must contain a \$DESCRIPTION file for you to enter PPE using this environment catalog.

User Action: Let PPE create a \$DESCRIPTION file for the environment catalog or enter PPE again, specifying a catalog that contains a \$DESCRIPTION file.

--ERROR SW 27-- Your terminal must support lines of at least 80 characters if you want to use the PPE.

Condition Identifier: SWE\$SCREEN_TOO_NARROW Code: SW 27

Description: PPE can only be used on terminals that support lines of at least 80 characters in width. Your terminal does not support 80 character lines, so you cannot use PPE on it.

User Action: Find a terminal that supports lines 80 or more characters wide and use it to execute PPE.

--ERROR SW 172-- The {text} screen is already in the screen stack. Access the screen by using the DISPLAY_SCREEN_STACK command.

Condition Identifier: SWE\$SCREEN_ALREADY_STACKED Code: SW 172

Description: You have already visited this screen, and it was placed on the screen stack; it has not been removed from the screen stack.

User Action: Use the screen stack to return to this screen or use the Back function until the screen is displayed again. The Back function removes the previous screen from the stack and displays it at the terminal.

--ERROR SW 173-- The processor {text} is not supported by the PPE.

Condition Identifier: SWE\$UNSUPPORTED_PROCESSOR Code: SW 173

Description: The processor that you specified is not supported by PPE. PPE supports the following processors:

- COBOL
- CYBIL
- FORTRAN (FORTRAN Version 1)
- VECTOR_FORTRAN (FORTRAN Version 2)

User Action: Specify one of the processors supported by PPE.

--ERROR SW 174-- The file \$DESCRIPTION in environment catalog {text} is busy, missing, or you are not permitted access. This condition must be repaired before catalogs at that level or higher can be used.

Condition Identifier: SWE\$UNREADABLE_DESCRIPTION_FILE Code: SW 174

Description: The file \$DESCRIPTION is missing or busy, or you are not permitted access to it; or the environment catalog containing \$DESCRIPTION is missing or busy, or you are not permitted access to it.

This message is displayed, for example, when you attempt to link your environment catalog to an environment catalog that does not exist.

User Action: To correct the condition that caused this message to be displayed, do one of the following:

- Check whether the environment catalog you named exists. If not, use the name of an existing environment catalog.
- Determine whether you have permission to the environment catalog and its files. If not, have the environment catalog's owner give you permission to link to it.
- Determine whether the \$DESCRIPTION file of the environment catalog you named is busy. If so, then wait until it is available for use.

--ERROR SW 185-- The level name {text} is not known to the PPE.

Condition Identifier: SWE\$UNKNOWN_LEVEL_NAME Code: SW 185

Description: The level name that you specified does not exist in your PPE catalog hierarchy.

User Action: Change the level name to one that exists in your PPE environment catalog hierarchy. If you do not know the names of the levels in your hierarchy, go to the Environment Description screen. The level names are listed in the Hierarchy level names field.

--ERROR SW 187-- {text} is not an existing catalog. You must specify an existing catalog, or no catalog at all.

Condition Identifier: SWE\$LINKED_TO_MUST_BE_CATALOG Code: SW 187

Description: You must link to an environment catalog to add your PPE catalog to a hierarchy. The owner of the catalog must provide the appropriate access permissions before you can link to it.

User Action: Type the name of an environment catalog in the field to link your PPE catalog to a hierarchy or leave the field blank.

--ERROR SW 188-- The link provided forms a loop in the catalog hierarchy because it leads to catalog {text}, which is linked to the current catalog. Such loops are not allowed.

Condition Identifier: SWE\$LOOP_IN_CATALOG_HIERARCHY Code: SW 188

Description: You cannot link your PPE environment catalog to itself or to any environment catalog below it in its PPE environment catalog hierarchy. For example, in the hierarchy shown below, the environment catalog named TOP cannot be linked to itself, to MIDDLE, or to BOTTOM.



Linking TOP to itself, MIDDLE, or BOTTOM would form a loop in the PPE environment catalog hierarchy. PPE does not allow such loops.

User Action: Link your environment catalog to an environment catalog that is not already in your hierarchy.

--WARNING SW 252-- The modification {text} does not exist. If you elect to create it now, the default parameter values will be determined by SCU, and not by PPE. Defaults specified on the Tailor Options screen, like the Author Name, will not be used.

Condition Identifier: SWE\$CREATE_MOD_DECISION Code: SW 252

Description: Each new deck must be associated with a modification, called the deck's creation modification.

The modification you have selected does not already exist. Therefore, PPE will create a new modification to associate with the new deck unless you change the modification name to one that already exists. The parameter values of the new modification will be determined by SCU defaults, not by the values specified on the Tailor Options screen.

User Action: Change the modification name to one that already exists or let PPE create a new modification using SCU defaults.

--ERROR SW 253-- The deck {text} already exists in the hierarchy level {text}.

Condition Identifier: SWE\$DECK_ALREADY_EXISTS Code: SW 253

Description: The name you specified for the new deck already belongs to a deck in your environment catalog hierarchy. The name of a new deck must be unique among all decks in your hierarchy.

User Action: Specify a deck name unique to your environment catalog hierarchy.

--ERROR SW 254-- You must specify a creation modification.

Condition Identifier: SWE\$MUST_SPECIFY_CREATION_MOD Code: SW 254

Description: Each new deck must be associated with a modification, called the deck's creation modification.

User Action: Enter the name of the deck's creation modification in the

Creation modification: _____

field at the top of the Deck Creation screen. The name can be that of an existing modification or a new modification. However, a new modification created in this fashion on the Deck Creation screen is created using SCU defaults, not using the information from the Tailor Options screen.

--ERROR SW 255-- The modification {text} exists in hierarchy level {text} but not in your working level, and thus is unavailable for use.

Condition Identifier: SWE\$UNAVAILABLE_MOD Code: SW 255

Description: The creation modification for the new deck is not at your level of the environment catalog hierarchy. Only a modification at your level of the hierarchy can be associated with the new deck.

User Action: Enter the name of a modification that is at your level of the environment catalog hierarchy or the name of a new modification. If you enter the name of a new modification, PPE creates the modification using SCU defaults, not the values from the Tailor Options screen.

--ERROR SW 273-- The catalog specified by the ENVIRONMENT_CATALOG parameter contains a subcatalog named SOURCE_LIBRARY which is expected to be a file.

Condition Identifier: SWE\$SOURCE_LIBRARY_IS_A_CATALOG Code: SW 273

Description: The catalog that you specified contains a sub-catalog named SOURCE_LIBRARY. A PPE environment catalog must contain a file named SOURCE_LIBRARY, which contains the source decks on which you work; therefore, the catalog that you specified cannot be used as a PPE environment catalog.

User Action: Specify a PPE environment catalog on the ENVIRONMENT_CATALOG parameter or remove or rename the SOURCE_LIBRARY sub-catalog in this catalog.

--ERROR SW 287-- The modification {text} already exists in the hierarchy level {text}.

Condition Identifier: SWE\$MOD_ALREADY_EXISTS Code: SW 287

Description: A modification with the name you specified already exists in your hierarchy. A PPE environment catalog hierarchy cannot have duplicate modification names. Therefore, you cannot create a new modification with the same name as another modification in your catalog hierarchy.

User Action: Change the name of the new modification to one different from any existing modification names.

--ERROR SW 289-- The modification {text} does not exist at the lowest level.

Condition Identifier: SWE\$MOD_NOT_AT_LOWEST_LEVEL Code: SW 289

Description: The current modification must exist at your level of the environment catalog hierarchy.

User Action: Change the current modification by doing one of the following:

- Type the name of a modification that exists at your level into the Current mod: _____ field.
- Create a new modification and type its name into the Current mod: _____ field.
- Extract a deck from a modification at a higher level in your environment catalog hierarchy. Then type the name of that deck's modification into the Current mod: _____ field.

--ERROR SW 462-- You must specify decks to extract by marking the desired decks, or by positioning the cursor on a deck.

Condition Identifier: SWE\$MUST_SPECIFY_EXTS_DECKS Code: SW 462

Description: You attempted to extract one or more decks from a higher level in your environment catalog hierarchy without identifying the decks to transmit.

User Action: To extract one or more decks, raise the display ceiling to the hierarchy level from which you want to extract and do either of the following:

- Place the cursor on a deck name, or
- Mark one or more deck names.

Then, to extract, press the ExtS function key.

--WARNING SW 545-- {text} has not been added to your job command list because it is not a known file. It is entry number {text} in your PPE command list.

Condition Identifier: SWE\$SETCL_ADD_UNKNOWN_NAME Code: SW 545

Description: You must provide PPE with the name of an existing file to add to the list entries.

User Action: Change the name of the file to the name of an existing file.

--ERROR SW 656-- The parameter list {text} already exists for this processor.

Condition Identifier: SWE\$DUPLICATE_PARAMETER_LIST Code: SW 656

Description: A parameter list with the same name already exists for this processor. The parameter list library does not allow duplicate parameter list names for the same processor.

User Action: Enter a unique name for the new parameter list.

--ERROR SW 661-- You must specify decks to transmit by marking the desired decks or by positioning the cursor over a deck.

Condition Identifier: SWE\$E_MUST_SPECIFY_TRAS_DECKS Code: SW 661

Description: You attempted to transmit one or more decks to the next higher level of your environment catalog hierarchy without identifying the decks to transmit.

User Action: To transmit one or more decks, do either of the following:

- Place the cursor on a deck name, or
- Mark one or more deck names.

Then, to transmit, press the TraS function key.

--ERROR SW 692-- You must specify the decks to build by marking the desired decks or by placing the cursor on a deck.

Condition Identifier: SWE\$E_MUST_SPECIFY_BUILD_DECKS Code: SW 692

Description: You attempted to build decks using the BuiD function without specifying which decks to build. You must specify which decks to build when you use the BuiD function or the BUILD_DECK command.

User Action: To specify which decks to build, do one of the following:

- Mark the deck to build
- Mark a range of decks to build
- Put the cursor on the deck to build

Then, to build the deck or decks indicated, use the BuiD function.

--ERROR TU 8010-- TDU TERMINATED WITH ERRORS.

Condition Identifier: TUE\$ERROR_TERMINATION Code: TU 8010

Description: This general message indicates that the terminal definition compiler found one or more errors in the input file. Examine the LIST file from the terminal definition compilation for details.

User Action: Correct the problem(s) and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8020-- CHARACTER VALUE MUST RANGE FROM 0 TO 255.

Condition Identifier: TUE\$INVALID_CHARACTER Code: TU 8020

Description: The values for the IN, OUT, or INOUT parameters must be ASCII characters. These may be entered as SCL strings, variables, or numeric ASCII values. The numeric value for a character must range from 0 to 255.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8030-- INVALID {text}.

Condition Identifier: TUE\$INVALID Code: TU 8030

Description: The terminal definition compiler did not expect to find the indicated text at the position where it was found.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8040-- INVALID TYPE--ONLY STRING, INTEGER, OR VARIABLE ALLOWED.

Condition Identifier: TUE\$INVALID_TYPE Code: TU 8040

Description: A boolean value or unrecognized characters appeared in a parameter value where only a character value (composed of an SCL string, ASCII value, and/or variable) is expected.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8050-- INVALID COMMUNICATIONS TYPE.

Condition Identifier: TUE\$INVALID_COMM Code: TU 8050

Description: The value for communications type may only be ASYNCH (asynchronous).

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8060-- INVALID CURSOR_ENCODING.

Condition Identifier: TUE\$INVALID_CURSOR Code: TU 8060

Description: The value for CURSOR_POS_ENCODING parameter may only be BINARY_CURSOR, CDC721_CURSOR, or ANSI_CURSOR.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8070-- CURSOR_BIAS OUT OF RANGE, MUST BE -255 TO 255.

Condition Identifier: TUE\$BIAS_OUT_OF_RANGE Code: TU 8070

Description: The value for cursor_bias is added to the row and column positions sent to the terminal. It must range from -255 to 255.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8080-- CONTINUATIONS EXCEED 256 CHARACTERS.

Condition Identifier: TUE\$CONTINUATION_OVERFLOW Code: TU 8080

Description: The length of an input line to the terminal definition compiler may not exceed 256 characters. This includes the accumulated lengths of continuation lines.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8090-- EXPECTING VERB OR VARIABLE, FOUND {text}.

Condition Identifier: TUE\$EXPECTING_VERB_VARIABLE Code: TU 8090

Description: Each line of the input file for DEFINE_TERMINAL must begin with a name which is either a parameter or a variable (blank lines and comments are also acceptable). The unrecognized text may be a misspelling, or a continuation indicator may be missing.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8100-- EMPTY INPUT FILE.

Condition Identifier: TUE\$EMPTY_FILE Code: TU 8100

Description: The input file supplied on the DEFINE_TERMINAL command was empty or nonexistent.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8110-- NO ROOM IN TABLE FOR {text}.

Condition Identifier: TUE\$NO_ROOM Code: TU 8110

Description: The terminal definition compiler has run out of space.

User Action: Try reducing the number of statements used (statements with null values can be removed), reducing the usage of variables, shortening names, etc. This error probably indicates a lack of virtual memory. You may require assistance from system personnel.

--ERROR TU 8120-- VERB {text} APPEARS TWICE.

Condition Identifier: TUE\$DUPLICATE_VERB Code: TU 8120

Description: The indicated statement has already been found in the input file.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8130-- INPUT SEQUENCE FOR {text} IS A SUPERSET OF A PREVIOUS ITEM.

Condition Identifier: TUE\$SUPERSET Code: TU 8130

Description: Input sequences (values for the IN and INOUT parameters) from terminals must be unique, or the computer would be unable to decide which sequence it had received. The first part of the sequence for the indicated parameter duplicates a sequence given with some previous parameter.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8140-- INPUT SEQUENCE FOR {text} IS A SUBSET OF A PREVIOUS ITEM.

Condition Identifier: TUE\$SUBSET Code: TU 8140

Description: Input sequences (values for the IN and INOUT parameters) from terminals must be unique, or the computer would be unable to decide which sequence it had received. The sequence for the indicated parameter duplicates the first part of a sequence given with some previous parameter.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8150-- VARIABLE {text} HAS NOT BEEN DECLARED.

Condition Identifier: TUE\$NOT_FOUND Code: TU 8150

Description: A name was found in a context which indicated that it must be a variable, but no variable has been defined with that name.

User Action: Check the spelling, and make sure that the variable was defined BEFORE it is used. Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8160-- TABLE OVERFLOW DURING OPTIMIZATION.

Condition Identifier: TUE\$OPTIMIZE_TABLE_FULL Code: TU 8160

Description: The terminal definition compiler has run out of space.

User Action: Try reducing the number of statements used (all statements with null values can be removed), reducing the usage of variables, shortening names, etc. This error probably indicates a lack of virtual memory. You may require assistance from system personnel.

--ERROR TU 8170-- INVALID VERB OR MISSING "=" IN VARIABLE ASSIGNMENT.

Condition Identifier: TUE\$INVALID_VERB_VARIABLE Code: TU 8170

Description: The name which begins this line of input for the DEFINE_TERMINAL commands is not a recognized parameter. If it is a variable to be assigned a value, it must be followed by an equals (=) sign.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8180-- DUPLICATE PARAMETERS, BOTH "IN" AND "INOUT".

Condition Identifier: TUE\$DUPLICATE_IN_INOUT Code: TU 8180

Description: The INOUT parameter combines the functions of the IN and the OUT parameters. You may not use INOUT with the same parameter as either IN or OUT.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8190-- DUPLICATE PARAMETERS, BOTH "OUT" AND "INOUT".

Condition Identifier: TUE\$DUPLICATE_OUT_INOUT Code: TU 8190

Description: The INOUT parameter combines the functions of the IN and the OUT parameters. You may not use INOUT with the same parameter as either IN or OUT.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8200-- TOO MANY SCREEN SIZES SPECIFIED, MAXIMUM 4.

Condition Identifier: TUE\$SCREEN_SIZE_OVERFLOW Code: TU 8200

Description: The terminal definition compiler supports from 1 to 4 screen sizes h for each terminal.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8210-- NUMBER OF ROWS MUST RANGE FROM 0 TO 255.

Condition Identifier: TUE\$SCREEN_ROW_OVERFLOW Code: TU 8210

Description: The terminal definition compiler cannot support a terminal with more than 255 rows.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8220-- NUMBER OF COLUMNS MUST RANGE FROM 0 TO 255.

Condition Identifier: TUE\$SCREEN_COL_OVERFLOW Code: TU 8220

Description: The terminal definition compiler cannot support a terminal with more than 255 columns.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8230-- INVALID "MOVE_PAST.." OR "CHAR_PAST.." TYPE.

Condition Identifier: TUE\$INVALID_CURSOR_BEHAVIOR Code: TU 8230

Description: The values for the MOVE_PAST.. or CHAR_PAST.. parameters may only be STOP_NEXT, SCROLL_NEXT, HOME_NEXT, WRAP_ADJACENT_NEXT, or WRAP_SAME_NEXT.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8240-- NAME IS REQUIRED.

Condition Identifier: TUE\$NAME_REQUIRED Code: TU 8240

Description: A value must be supplied for the MODEL_NAME parameter.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8250-- NAME MUST BE 1 TO 25 CHARACTERS.

Condition Identifier: TUE\$NAME_TOO_LONG Code: TU 8250

Description: The value for the MODEL_NAME parameter must have from 1 to 25 characters.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8260-- "OUT" REQUIRED FOR SET_SIZE.

Condition Identifier: TUE\$SCREEN_OUT_REQUIRED Code: TU 8260

Description: The SET_SIZE parameter defines the possible screen sizes and the character sequences necessary to activate them. You must supply the OUT parameter with this parameter.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8270-- INVALID NAME--MAY ONLY BE ALPHABETIC AND NUMERIC CHARACTERS, OR UNDERLINE.

Condition Identifier: TUE\$INVALID_NAME Code: TU 8270

Description: The value for the MODEL_NAME parameter must consist of alphabetic and numeric characters only.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8280-- CURSOR_POS_COLUMN_LENGTH OR CURSOR_POS_ROW_LENGTH MUST BE 0 TO 7.

Condition Identifier: TUE\$CURSOR_LEN_OUT_OF_RANGE Code: TU 8280

Description: The number of characters sent to the terminal for a row or column position ranges from 1 to 7. Enter 0 if this number is not fixed.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8290-- INPUT SEQUENCE FOR {text} IS A DUPLICATE OF A PREVIOUS ITEM.

Condition Identifier: TUE\$DUPLICATE_INPUT Code: TU 8290

Description: Input sequences (values for the IN and INOUT parameters) from terminals must be unique, or the computer would be unable to decide which sequence it had received. The sequence for the indicated parameter duplicates a sequence given with some previous parameter.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8310-- APPLICATION STRING NAME MUST BE 1 TO 7 CHARACTERS.

Condition Identifier: TUE\$APPSTR_NAME_TOO_LONG Code: TU 8310

Description: Application string name must conform to the NOS/VE NAME convention of 1 to 31 alphabetic characters, digits, or _ \$ # @. First character must not be a digit.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8320-- FUNCTION_KEY_LEAVES_MARK MUST INDICATE 0 TO 7 CHARACTERS OF BLOTCH.

Condition Identifier: TUE\$FUNCTION_KEY_MARK_RANGE Code: TU 8320

Description: When a function key is pressed on some terminals, it leaves a number of garbage characters on the screen. If so, indicate how many characters are left. If not, indicate 0.

User Action: Correct the problem and re-enter the DEFINE_TERMINAL command.

--ERROR TU 8330-- THE NUMBER GIVEN FOR PROGRAMMABLE TAB STOPS MUST RANGE FROM 0 TO 255.

Condition Identifier: TUE\$PROG_TAB_STOP_RANGE Code: TU 8330

Description: The number of programmable tab stops an application can set for a particular terminal ranges from 0 to 255.

User Action: Correct the number of tab stops and reenter the DEFINE_TERMINAL command.

--ERROR TU 8340-- THE NUMBER OF FIXED TAB STOPS THAT MAY BE GIVEN IS FROM 0 TO 255.

Condition Identifier: TUE\$FIXED_TAB_STOP_RANGE Code: TU 8340

Description: The number of fixed tab stops supported for a terminal ranges from 0 to 255.

User Action: Correct the number of tab stops and reenter the DEFINE_TERMINAL command.

--ERROR TU 8350-- THE VALUE GIVEN FOR TAB STOP POSITIONS MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$TAB_STOP_POSITION_RANGE Code: TU 8350

Description: A tab stop position value from 1 to 255 must be specified for each tab stop.

User Action: Correct the tab stop positions and reenter the DEFINE_TERMINAL command.

--ERROR TU 8360-- THE VALUE GIVEN FOR CHARACTER POSITIONS ACCURACY MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$CHARACTER_POSITIONS_RANGE Code: TU 8360

Description: The SET_SIZE parameter CHARACTER_POSITIONS specifies the possible cursor character positions for each pick/locate operation. The character position value must range from 1 to 255.

User Action: Correct the parameter and reenter the DEFINE_TERMINAL command.

--ERROR TU 8370-- THE VALUE GIVEN FOR LINE POSITIONS ACCURACY MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$LINE_POSITIONS_RANGE Code: TU 8370

Description: The SET_SIZE parameter LINE_POSITIONS specifies the possible cursor line positions for each pick/locate operation. The line-position value must be from 1 to 255.

User Action: Correct the parameter and reenter the DEFINE_TERMINAL command.

--ERROR TU 8380-- THE VALUE GIVEN FOR STARTING POSITION MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$STARTING_POSITION_VALUE Code: TU 8380

Description: The SET_SIZE parameters CHARACTER_SPECIFICATION and LINE_SPECIFICATION must have a starting_position value that ranges from 1 to 255.

User Action: Correct the parameter(s) and reenter the DEFINE_TERMINAL command.

--ERROR TU 8390-- THE VALUE GIVEN FOR ENDING POSITION MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$ENDING_POSITION_VALUE Code: TU 8390

Description: The SET_SIZE parameters CHARACTER_SPECIFICATION and LINE_SPECIFICATION must have an ending_position value that ranges from 1 to 255.

User Action: Correct the parameter(s) and reenter the DEFINE_TERMINAL command.

--ERROR TU 8400-- THE VALUE GIVEN FOR INCREMENT MUST RANGE FROM 1 TO 255.

Condition Identifier: TUE\$INCREMENT_INVALID Code: TU 8400

Description: The SET_SIZE parameters CHARACTER_SPECIFICATION and LINE_SPECIFICATION must have an increment value that ranges from 1 to 255.

User Action: Correct the parameter(s) and reenter the DEFINE_TERMINAL command.

--ERROR XM 0-- No data received from Sender. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$NO_DATA_FROM_SENDER Code: XM 0

Description: The NOS/VE XMODEM software application did not receive any data from the micro's XMODEM software after 10 ten-second timeouts. The host terminated the transfer.

User Action: Check that both the host and micro support the XMODEM protocol extension being used. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 2-- Data received is out of sequence. Packet sequence number = {text}. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$DATA_OUT_OF_SEQUENCE Code: XM 2

Description: The NOS/VE XMODEM software application received a packet with a sequence number that is neither the expected sequence number nor the last sequence number and has terminated the file transfer.

User Action: Try the transfer again. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application should be given to a site analyst for analysis by specifying the TRACE_FILE parameter on the host command.

--ERROR XM 3-- Transfer terminated unexpectedly by Sender.

Condition Identifier: XME\$SENDER_TERMINATED_TRANSFER Code: XM 3

Description: The NOS/VE XMODEM software application received a CAN (cancel) character from the micro's XMODEM software.

User Action: Check the micro's XMODEM software to determine why it aborted the file transfer.

--ERROR XM 4-- Initial startup character not received from Receiver. Transfer terminated abnormally by Sender.

Condition Identifier: XME\$NO_STARTUP_CHARACTER Code: XM 4

Description: The NOS/VE XMODEM software application did not receive a startup character from the micro's XMODEM software after 10 ten-second timeouts. The host terminated the transfer.

User Action: Check that the micro's XMODEM software package has been properly setup and started. Check that both the host and micro support the XMODEM protocol extension being used. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 5-- Acknowledgement not received from Receiver. Transfer terminated abnormally by Sender.

Condition Identifier: XME\$NO_ACK_FROM_RECEIVER Code: XM 5

Description: The NOS/VE XMODEM software application did not receive an acknowledgement from the micro's XMODEM software after 10 ten-second timeouts for the last packet or EOT character transmitted by the host. The transfer was terminated by the host. Note that after each ten-second timeout, the packet had been retransmitted to the micro.

User Action: Check the micro's XMODEM software to determine why the micro does not respond. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 6-- Transfer terminated unexpectedly by Receiver.

Condition Identifier: XME\$RECEIVER_TERMINATED_XFER Code: XM 6

Description: The NOS/VE XMODEM software application received a CAN (cancel) character from the micro's XMODEM software.

User Action: Check the micro's XMODEM software to determine why it aborted the file transfer.

--ERROR XM 7-- File {text} is empty. Transfer terminated abnormally by Sender.

Condition Identifier: XME\$FILE_EMPTY Code: XM 7

Description: A file requested to be sent to the micro contains no data (i.e. an empty file). The NOS/VE XMODEM software application terminated the file transfer.

User Action: The host command probably has a misspelled file name. Reissue the command with a valid file name for a file which contains data.

--ERROR XM 8-- File {text} is not readable. Transfer terminated abnormally by Sender.

Condition Identifier: XME\$FILE_NOT_READABLE Code: XM 8

Description: A file requested to be sent to the micro does not have write permission for the user. The NOS/VE XMODEM software application terminated the file transfer.

User Action: Obtain file permission to allow read access to the file and retry the file transfer.

--ERROR XM 9-- End Of Transmission character not acknowledged by receiver. Transfer terminated abnormally by Sender.

Condition Identifier: XME\$EOT_NOT_ACKED Code: XM 9

Description: The NOS/VE XMODEM software application timed out while waiting for an acknowledgment from the micro to the EOT character it transmitted. The host terminated the file transfer.

User Action: Check the micro's XMODEM software to determine why the micro did not respond. In addition, if the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 10-- Receiver detected data transmission error. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$G_OPTION_DATA_ERROR Code: XM 10

Description: The NOS/VE XMODEM software application timed out waiting for data from the micro during a G-Option file transfer. The host terminated the file transfer on the first ten-second timeout.

User Action: Check the micro's XMODEM software to determine why the micro did not respond. In addition, if the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 11-- Receiver detected data transmission error. First character of data received is an invalid response character. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$INVALID_RESPONSE_CHARACTER Code: XM 11

Description: The NOS/VE XMODEM software application received a packet that did not contain a valid start-of-header (SOH/STX) character from the micro during a G-Option file transfer. The host terminated the transfer.

User Action: Use the G-Option XMODEM protocol extension only on high speed, error free communications. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying 1 the TRACE_FILE parameter on the host command.

--ERROR XM 12-- Receiver detected data transmission error. Received error checking does not match calculated error checking for packet sequence number = {text}. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$ERROR_CHECKING_ERROR Code: XM 12

Description: The NOS/VE XMODEM software application received a packet from the micro during a G-Option file transfer for which the error checking value (checksum/crc) does not match the calculated error checking value. The host terminated the transfer.

User Action: Use the G-Option XMODEM protocol extension only on high speed, error free communications. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 13-- Receiver detected data transmission error. Received one's complement value for packet sequence number = {text} is incorrect. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$COMPLEMENT_ERROR Code: XM 13

Description: The NOS/VE XMODEM software application received a packet from the micro during a G-Option file transfer for which the sequence number does not match the ones complement sequence number. The host terminated the transfer.

User Action: Use the G-Option XMODEM protocol extension only on high speed, error free communications. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying the TRACE_FILE parameter on the host command.

--ERROR XM 14-- Receiver detected data transmission error. A duplicate packet for packet sequence number = {text} has been received. Transfer terminated abnormally by Receiver.

Condition Identifier: XME\$DUPLICATE_PACKET Code: XM 14

Description: The NOS/VE XMODEM software application received a packet from the micro during a G-Option file transfer for which the packet sequence number is the same as the previously received packet. The host terminated the transfer.

User Action: Use the G-Option XMODEM protocol extension only on high speed, error free communications. If the error can be replicated, a trace of the file transfer showing a hexadecimal dump of every protocol block sent and received by the host application can be given to a site analyst for analysis by rerunning the file transfer and specifying 1 the TRACE_FILE parameter on the host command.

--ERROR XM 15-- The 1024 byte data packet was not ACKnowledged by the Receiver. May be possible data transmission errors or the Receiver does not support Ymodem (1024 byte data packets). Transfer terminated abnormally by Sender.

Condition Identifier: XME\$LARGE_PACKET_NOT_ACKED Code: XM 15

Description: The NOS/VE XMODEM software application received a repeated negative acknowledgement from the micro for a packet containing an STX as the first character and 1024-byte data field. After 10 retransmissions, the host terminated the transfer.

User Action: Check the micro XMODEM software to be sure that it supports the XMODEM-1K protocol extension. If it does not, it should retry the transfer using small packet size.

Related Manuals

A

Ordering Printed Manuals	A-1
Accessing Online Manuals	A-1

Related Manuals

A

All NOS/VE manuals and related hardware manuals are listed in table A-1. If your site has installed the online manuals, you can find an abstract for each NOS/VE manual in the online System Information manual. To access this manual, enter:

```
/help manual=nos_ve
```

Ordering Printed Manuals

To order a printed Control Data manual, send an order form to:

Control Data
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103-2495

To obtain an order form or to get more information about ordering Control Data manuals, write to the above address or call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

Accessing Online Manuals

To access the online version of a printed manual, log in to NOS/VE and enter the online title on the HELP command (table A-1 supplies the online titles). For example, to see the NOS/VE Commands and Functions manual, enter:

```
/help manual=scl
```

or, because SCL is the default for the MANUAL parameter, simply enter

```
/help
```

An online Examples manual contains examples that reside in printed manuals. From within the online Examples manual, you can copy, print, and execute the examples it contains. To access this manual, enter:

```
/help manual=examples
```

When EXAMPLES is listed in the Online Manuals column in table A-1, that manual is represented in the online Examples manual.

Table A-1. Related Manuals

Manual Title	Publication Number	Online Manuals¹
NOS/VE Site Manuals:		
CYBER 930 Computer System Guide to Operations Usage	60469560	
CYBER Initialization Package (CIP) Reference Manual	60457180	
Desktop/VE Host Utilities Usage	60463918	
MAINTAIN_MAIL (Version 1) ² Usage		MAIM
NOS/VE Accounting Analysis System Usage	60463923	
NOS/VE Accounting and Validation Utilities for Dual State Usage	60458910	
NOS/VE File Server for STORNET and ESM-II Usage	60000190	
NOS/VE LCN Configuration and Network Management Usage	60463917	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

2. To access this manual, you must be the administrator for Mail/VE Version 1.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Site Manuals (Continued):		
NOS/VE Network Management Usage	60463916	
NOS/VE Operations Usage	60463914	
NOS/VE System Performance and Maintenance Volume 1: Performance Usage	60463915	
NOS/VE System Performance and Maintenance Volume 2: Maintenance Usage	60463925	
NOS/VE User Validation Usage	60464513	
NOS/VE User Manuals:		
EDIT_CATALOG Usage		EDIT_CATALOG
EDIT_CATALOG for NOS/VE Summary	60487719	
Introduction to NOS/VE Tutorial	60464012	EXAMPLES
NOS/VE Advanced File Management Tutorial	60486412	AFM_T

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
NOS/VE User Manuals (Continued):		
NOS/VE Advanced File Management Usage	60486413	AFM
NOS/VE Advanced File Management Summary	60486419	
NOS/VE Commands and Functions Quick Reference	60464018	SCL
NOS/VE File Editor Tutorial/Usage	60464015	EXAMPLES
NOS/VE Object Code Management Usage	60464413	OCM and EXAMPLES
NOS/VE Screen Formatting Usage	60488813	EXAMPLES
NOS/VE Source Code Management Usage	60464313	SCM and EXAMPLES
NOS/VE System Usage	60464014	EXAMPLES
NOS/VE Terminal Definition Usage	60464016	
Screen Design Facility for NOS/VE Usage	60488613	SDF

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
NOS/VE User Manuals (Continued):		
Screen Design Facility/Data Management Usage	60488618	
CYBIL Manuals:		
CYBIL for NOS/VE File Management Usage	60464114	EXAMPLES
CYBIL for NOS/VE Keyed-File and Sort/Merge Interfaces Usage	60464117	EXAMPLES
CYBIL for NOS/VE Language Definition Usage	60464113	CYBIL and EXAMPLES
CYBIL for NOS/VE Sequential and Byte-Addressable Files Usage	60464116	EXAMPLES
CYBIL for NOS/VE System Interface Usage	60464115	EXAMPLES

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
FORTRAN Manuals:		
FORTRAN Version 1 for NOS/VE Language Definition Usage	60485913	EXAMPLES
FORTRAN Version 1 for NOS/VE Quick Reference		FORTRAN
FORTRAN Version 2 for NOS/VE Language Definition Usage	60487113	EXAMPLES
FORTRAN Version 2 for NOS/VE Quick Reference		VFORTRAN
FORTRAN for NOS/VE Tutorial	60485912	FORTRAN_T
FORTRAN for NOS/VE Topics for FORTRAN Programmers Usage	60485916	
FORTRAN for NOS/VE Summary	60485919	
COBOL Manuals:		
COBOL for NOS/VE Summary	60486019	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
COBOL Manuals (Continued):		
COBOL for NOS/VE Tutorial	60486012	COBOL_T
COBOL for NOS/VE Usage	60486013	COBOL and EXAMPLES
Other Compiler Manuals:		
ADA for NOS/VE Usage	60498113	ADA
ADA for NOS/VE Reference Manual	60498118	EXAMPLES
APL for NOS/VE File Utilities Usage	60485814	
APL for NOS/VE Language Definition Usage	60485813	
BASIC for NOS/VE Summary Card	60486319	
BASIC for NOS/VE Usage	60486313	BASIC
LISP for NOS/VE Usage Supplement	60486213	
Pascal for NOS/VE Summary Card	60485619	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Other Compiler Manuals (Continued):		
Pascal for NOS/VE Usage	60485618	PASCAL and EXAMPLES
Prolog for NOS/VE Quick Reference	60486718	PROLOG
Prolog for NOS/VE Usage	60486713	
VX/VE Manuals:		
C/VE for NOS/VE Quick Reference		C
C/VE for NOS/VE Usage	60469830	
DWB/VX Introduction and User Reference Tutorial/Usage	60469890	
DWB/VX Macro Packages Guide Usage	60469910	
DWB/VX Preprocessors Guide Usage	60469920	
DWB/VX Text Formatters Guide Usage	60469900	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
VX/VE Manuals (Continued):		
VX/VE Administrator Guide and Reference Tutorial/Usage	60469770	
VX/VE An Introduction for UNIX Users Tutorial/Usage	60469980	
VX/VE Programmer Guide Tutorial	60469790	
VX/VE Programmer Reference Usage	60469820	
VX/VE Support Tools Guide Tutorial	60469800	
VX/VE User Guide Tutorial	60469780	
VX/VE User Reference Usage	60469810	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Data Management Manuals:		
DM Command Procedures Reference Manual	60487905	
DM Concepts and Facilities Manual	60487900	
DM Error Message Summary for DM on CDC NOS/VE	60487906	
DM Fundamental Query and Manipulation Manual	60487903	
DM Report Writer Reference Manual	60487904	
DM System Administrator's Reference Manual for DM on CDC NOS/VE	60487902	
DM Utilities Reference Manual for DM on CDC NOS/VE	60487901	
Information Management Manuals:		
IM/Control for NOS/VE Quick Reference	L60488918	CONTROL
IM/Control for NOS/VE Usage	60488913	
IM/Fast for NOS/VE Administration Usage	60487513	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Information Management Manuals (Continued):		
IM/Fast for NOS/VE Programming Usage	60487514	
IM/Quick for NOS/VE Tutorial	60485712	
IM/Quick for NOS/VE Summary	60485714	
IM/Quick for NOS/VE Online Help		QUICK
CDCNET Manuals:		
CDCNET Access Guide		CDCNET_ACCESS
CDCNET Batch Device User Guide		CDCNET_BATCH
CDCNET Command Quick Reference	60000020	
CDCNET Conceptual Overview	60461540	
CDCNET Configuration and Site Administration Guide	60461550	
CDCNET DI Dump Analyzer		ANACD
CDCNET Diagnostic Messages	60461600	CDCNET_MSGS

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
CDCNET Manuals (Continued):		
CDCNET Network Analysis	60461590	
CDCNET Network Configuration Utility		NETCU
CDCNET Network Configuration Utility Summary Card	60000269	
CDCNET Network Operations	60461520	
CDCNET Network Performance Analyzer	60461510	
CDCNET Product Descriptions	60460590	
CDCNET Systems Programmer's Reference Manual Volume 1 Base System Software	60462410	
CDCNET Systems Programmer's Reference Manual Volume 2 Network Management Entities and Layer Interfaces	60462420	
CDCNET Systems Programmer's Reference Manual Volume 3 Network Protocols	60462430	
CDCNET Terminal Interface Usage	60463850	
CDCNET TCP/IP Usage	60060214	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Migration Manuals:		
Migration from IBM to NOS/VE Tutorial/Usage	60489507	
Migration from NOS to NOS/VE Tutorial/Usage	60489503	
Migration from NOS to NOS/VE Standalone Tutorial/Usage	60489504	
Migration from NOS/BE to NOS/VE Tutorial/Usage	60489505	
Migration from NOS/BE to NOS/VE Standalone Tutorial/Usage	60489506	
Migration from VAX/VMS to NOS/VE Tutorial/Usage	60489508	
Miscellaneous Manuals:		
Applications Directory	60455370	
Control Data CONNECT User's Guide	60462560	
Control Data CONNECT Plus for the IBM Personal Computer (Version 1.0) User's Guide	60000388	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Miscellaneous Manuals (Continued):		
Debug for NOS/VE Quick Reference		DEBUG
Debug for NOS/VE Usage	60488213	
Desktop/VE for Macintosh Usage	60464503	
NOS/VE Diagnostic Messages Usage	60464613	MESSAGES
Mail/VE (Version 1) Summary Card	60464519	
Mail/VE (Version 1) Usage		MAIL_VE
Mail/VE Version 2 Administration	60464515	MAILVE_ ADMINISTRA- TION
Mail/VE Version 2 Usage	60464514	MAILVE_V2
Math Library for NOS/VE Usage	60486513	
NOS/VE Examples Usage		EXAMPLES

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Miscellaneous Manuals (Continued):		
NOS/VE Online Manuals Systems	60488403	TOPICS_ CONTEXT
NOS/VE System Information		NOS_VE
Programming Environment for NOS/VE Usage		ENVIRON- MENT
Programming Environment for NOS/VE Summary	60486819	
Professional Programming Environment for NOS/VE Quick Reference		PPE
Professional Programming Environment for NOS/VE Usage	60486613	
Remote Host Facility Usage	60460620	
Hardware Manuals:		
CYBER 170 Computer Systems Models 825, 835, and 855 General Description Hardware Reference	60459960	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table A-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Hardware Manuals (Continued):		
CYBER 170 Computer Systems, Models 815, 825, 835, 845, and 855 CYBER 180 Models 810, 830, 835, 840, 845, 850, 855, and 860 Codes Booklet	60458100	
CYBER 170 Computer Systems, Models 815, 825, 835, 845, and 855 CYBER 180 Models 810, 830, 835, 840, 845, 850, 855, and 860 Maintenance Register Codes Booklet	60458110	
HPA/VE Reference	60461930	
Virtual State Volume II Hardware Reference	60458890	
7021-31/32 Advanced Tape Subsystem Reference	60449600	
7221-1 Intelligent Small Magnetic Tape Subsystem Reference	60461090	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

Please fold on dotted line;
seal edges with tape only.

FOLD

FOLD

FOLD

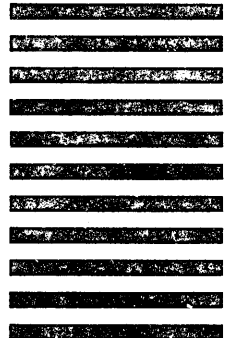


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983



We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

How do you use this manual?

- Manager
- Systems analyst or programmer
- Applications programmer
- Operator
- Other _____

- As an overview
- To learn the product or system
- For comprehensive reference
- For quick look-up
- Other _____

What programming languages do you use? _____

How do you like this manual? Answer the questions that apply.

- | Yes | Somewhat | No | |
|--------------------------|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Does it tell you what you need to know about the topic? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the technical information accurate? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is it easy to understand? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the order of topics logical? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Can you easily find what you want? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Are there enough examples? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Are the examples helpful? (<input type="checkbox"/> Too simple? <input type="checkbox"/> Too complex?) |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Do the illustrations help you? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the manual easy to read (print size, page layout, and so on)? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Do you use this manual frequently? |

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply:

Name _____

Company _____

Address _____

Date _____

Phone _____

Please send program listing and output if applicable to your comment.

