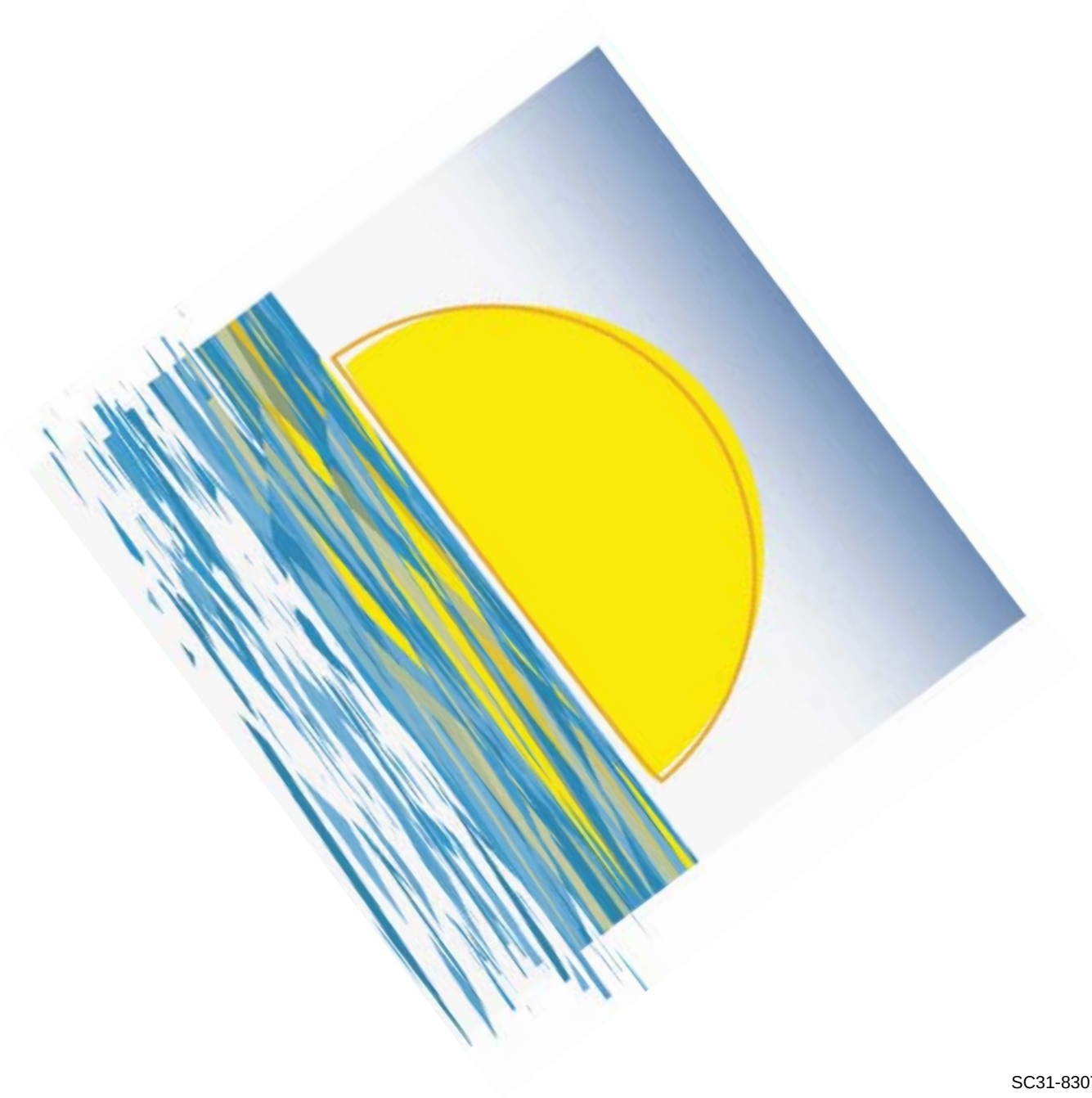
OS/390 TCP/IP OpenEdition

Messages and Codes



SC31-8307-00

OS/390 TCP/IP OpenEdition

Messages and Codes

SC31-8307-00

Note

Before using this information and the product it supports, be sure to read the general information under Appendix B, “Notices” on page 239.

First Edition (June 1997)

This edition applies to OS/390 (5645-001) and OS/390 TCP/IP OpenEdition. See the “Summary of Changes” for a description of the changes made in this edition. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be at the back of this publication. If the form has been removed, you may send your comments to the following address:

International Business Machines Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
USA

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):	1-800-227-5088
Internet e-mail:	usib2hpd@vnet.ibm.com
World Wide Web:	http://www.s390.ibm.com/os390
IBMLink:	CIBMORCF at RALVM13
IBM Mail Exchange:	USIB2HPD at IBMMAIL

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About This Book

This book provides information about the messages and codes that occur in OS/390 TCP/IP OpenEdition. This book can also help you to determine whether a specific problem is a result of the MVS TCP/IP implementation.

For information about how to set up, initialize, and customize your TCP/IP for MVS system, see *OS/390 TCP/IP OpenEdition Configuration Guide* and *OS/390 TCP/IP OpenEdition Programmer's Reference*. For information about how to use the applications on your TCP/IP for MVS system, see *OS/390 TCP/IP OpenEdition User's Guide*.

For comments and suggestions about this book, use the Reader's Comment Form located at the back of this book. This form provides instructions on submitting your comments by mail, by fax, or by electronic mail.

IBM TCP/IP V3R2 for MVS is an integral part of the OS/390 family of products. For an overview and mapping of the documentation available for OS/390, see the *OS/390 Information Roadmap*.

Who Should Use This Book

This book is intended for use by TCP/IP operators, system programmers, and users for diagnosing problems. You should also use this book to:

- Analyze a problem
- Classify the problem as a specific type
- Describe the problem to the IBM Software Support Center

Before using this book, you should be familiar with TCP/IP and the protocol commands. In addition, OS/390 TCP/IP OpenEdition should already be installed and customized for your network.

Terms and Abbreviations Used in This Book

This section describes various terms that are used throughout the book.

How the Term "internet" Is Used in This Book

In this book, an internet is a logical collection of networks supported by routers, gateways, bridges, hosts, and various layers of protocols, which permit the network to function as a large, virtual network.

Note: The term "internet" is used as a generic term for a TCP/IP network, and should not be confused with the Internet, which consists of large national backbone networks (such as MILNET, NSFnet, and CREN) and a myriad of regional and local campus networks worldwide.

How the Book Distinguishes "TCPIP" from "TCP/IP"

The abbreviation TCPIP is used to refer to the specific address space on which the Transmission Control Protocol/Internet Protocol product resides (for example, "TCPIP continues..."). The abbreviation TCP/IP refers to the product itself, as in *OS/390 TCP/IP OpenEdition Configuration Guide*.

Throughout this book, the abbreviation MVS refers to the following IBM products:

- IBM Multiple Virtual Storage/System Product Version 2 Release 1.3 (MVS/XA), or later
- IBM Multiple Virtual Storage/System Product Version 3 Release 1.0 (MVS/ESA), or later
- IBM Multiple Virtual Storage/System Product Version 4.1 (MVS/SP), or later

The following abbreviations are also used in this book:

ES/9000	Enterprise System/9000 processor.
ES/9370	Enterprise System/9370 processor.
NCS**	Network Computing System,** which is the Apollo** implementation of remote procedure calls.

RPC	Sun** Microsystems' implementation of remote procedure calls.
SQL	IBM Structured Query Language.
SQL/DS	IBM Structured Query Language/Data Systems Version 2 Release 2 or later.

Within the TCP/IP environment, you should also be familiar with the following terms:

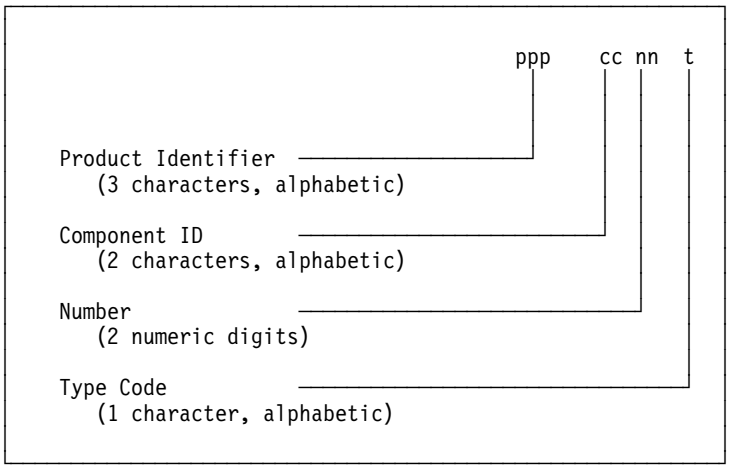
Term	Description
Data set	The basic unit of data storage for MVS. Unless otherwise specified, the use of this term indicates that the MVS host storing your local data set is your local host system.
Local host	In an internet, any computer to which an end user or a functional unit is connected without the use of the internet.
Remote host	In an internet, any host on the network that requires a physical link to interconnect with the network.
User, local user	Either servers or clients (address spaces) on the local MVS system.

The variable *hlq* is used in this book to represent the high-level qualifier of a data set name. The default high-level qualifier for your installation was set when the product was installed and customized. Because TCP/IP for MVS uses both implicit and explicit data set allocation and provides you with methods of overriding the default, this variable can have many possible values. The *OS/390 TCP/IP OpenEdition Configuration Guide* and the *OS/390 TCP/IP OpenEdition Planning and Release Guide* provide more details about data set names and the use of high-level qualifiers in this release. When in doubt, check with your system administrator to determine the value of *hlq* for a particular data set.

The IBM TCP/IP Message Standard

This section describes message number conventions used in this manual.

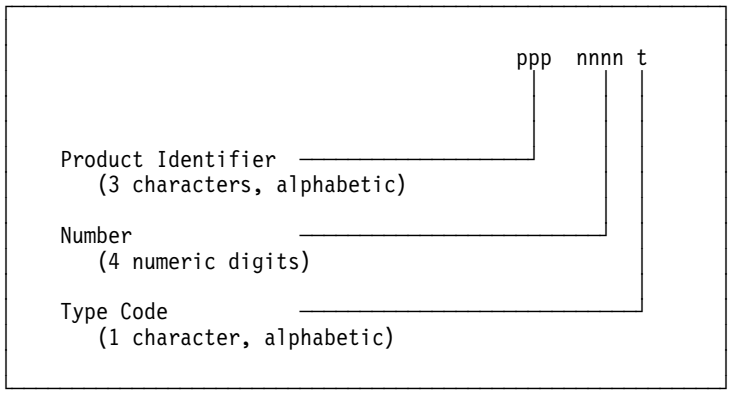
The following diagram explains the older message number convention that was used for the TCP/IP for OpenEdition MVS applications. Some messages with EZY prefixes have a 4 digit Number and no Component ID.



The following table describes the meanings for the type code characters:

Type Code	Severity
E or S	A recoverable error occurred. Follow the instructions given in the book to fix the problem.
F,T, or U	An irrecoverable error that terminated the program occurred. You might need to contact the IBM Software Support Center.
I	Informational message.
N	Nondisplay. The message text, but not the message ID, appeared.
W	Warning. A condition that might subsequently cause an error has occurred.
X	Concatenation. This message is adjoined to one or more messages.

The following diagram explains the message number convention currently in use:



The ***product identifiers*** (ppp) for TCP/IP are EZY and EZZ. The ***number*** (nnnn) indicates a unique 4-digit numeric value assigned to the message by product. The ***type*** (t) indicates the action assigned to the message. The message ID is followed by a single space and the message text.

For example:

EZZ0902I Use a host name after HOST option.

The following list identifies and describes the alphabetic character at the end of the message ID.

Letter	Meaning
A	Immediate Action required.
E	Eventual Action required.
D	Immediate Decision required.
I	Informational.

The following tables show the relationship between the TCP/IP components and their corresponding message IDs ranges:

OS/390 TCP/IP OpenEdition Component Identifiers and Message Ranges

EZY Messages

OS/390 TCP/IP OpenEdition includes messages for previously shipped OE applications. These EZY messages use the component identifiers listed in the following table.

Table 1 (Page 1 of 2). EZY Message Component Identifiers

EZY Component ID	Component Name
FS	FTP Server
FT	FTP Server
RC	orexec Client
RD	rexecd Daemon
RG	RPCGEN
RS	RSHD Client
TE	Telnet Server - Executive
TO	Telnet Server - Other
TS	Telnet Server - State
TU	Telnet Server - Utility
TY	Telnet - System
XM	OSF/Motif

Table 1 (Page 2 of 2). EZY Message Component Identifiers	
EZY Component ID	**Component Name**
XN	OSF/Motif
XO	OSF/Motif
XP	OSF/Motif
XQ	OSF/Motif
XR	OSF/Motif
XU	OSF/Motif
XW	X Window System

EZZ Messages

The following table describes the message ranges for the EZZ messages used by OS/390 TCP/IP OpenEdition.

Table 2. EZZ Message Ranges	
EZZ Message ID Range	**Component**
EZZ0050I–EZZ0348I	Configuration
EZZ2351I–EZZ2382I	onetstat
EZZ3100I–EZZ3119I	oping
EZZ3200I–EZZ3220I	SNMP SubAgent
EZZ3250I–EZZ3255I	Operator Command Related
EZZ3300I–EZZ3327I	osnmp
EZZ4200I–EZZ4216I	Initialization
EZZ4300I–EZZ4315I	If Layer
EZZ4820I–EZZ4998I	orouted
EZZ6201I–EZZ6274I	osnmpd
EZZ6354I–EZZ6376I	otracert

Online Message Facility

The Online Message Facility is an OS/2 program that provides OS/2 users with online access to information from *OS/390 TCP/IP OpenEdition Messages and Codes*. The facility helps network operators and system programmers operate and diagnose problems without interrupting those tasks.

You can retrieve the descriptions of TCP/IP host messages by double-clicking on the message ID in a Communications Manager emulator window. The Online Message Facility extracts the information you selected, builds a series of search arguments, then passes those arguments to BookManager READ/2. The BookManager program searches for the information you selected and opens the *OS/390 TCP/IP OpenEdition Messages and Codes* book to the description of that message. In addition, you can take advantage of all the functions of BookManager READ/2 once it is invoked by the facility.

If you are not using Communications Manager, you still can retrieve TCP/IP host message descriptions by clicking on the Online Message ICON. If you click on SEARCH and then MESSAGE, a menu appears allowing you to enter the message ID. After entering the message ID, a BookManager search is invoked. The BookManager program searches for the information you selected and opens the *OS/390 TCP/IP OpenEdition Messages and Codes* book to the description of that message. In addition, you can take advantage of all the functions of BookManager READ/2 once it is invoked by the facility.

The Online Message Facility is shipped on the *IBM Networking Systems Softcopy Collection Kit* CD-ROMs. Information on planning for, installing, and using the facility is in *Using the Online Message Facility*, which is also available on the CD-ROM.

You must have a CD-ROM drive to copy the facility to your workstation.

Crossbook-Linking Enhancement

Enhancements have been made to the OS/390 TCP/IP OpenEdition softcopy books. These enhancements take advantage of BookManager's linking functions to improve the usability and retrievability of the TCP/IP softcopy library. These functions allow you to move between related pieces of information. Links can go from one point to another within the same softcopy book or can go from one softcopy book to another.

Enabled links are often highlighted when the book is viewed on the display screen. Simply press ENTER on a highlighted word, phrase, reference, command, message ID, or configuration statement and you will automatically be linked to additional information.

Note: Links between books are not effective unless the softcopy books are registered in a public library, and users have access to them.

Where to Find Related Information

For more information about the HYPERchannel** feature, see *HYPERchannel A220 Processor Adapter* 4290007, Network Systems Corporation.

Where to Find Related Information on the Internet

You may find the following information helpful.

For current updates to the IBM TCP/IP V3R2 for MVS documentation described in “Bibliography” on page 243, check out the TCP/IP for MVS home page:

`http://www.networking.ibm.com/tcm/tcmprod.html`

To keep in close touch with OS/390, we suggest you look at the OS/390 home page:

`http://www.s390.ibm.com/os390`

To keep abreast of new products and technologies from IBM Networking, take a look at the IBM Networking home page:

`http://www.networking.ibm.com/`

The IBM Networking Software Glossary is now available in HTML format as well as PDF. You can access it directly at the following URL:

`http://www.networking.ibm.com/nsg/nsggls.htm`

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Summary of Changes

Summary of Changes
for SC31-8307-00
OS/390 TCP/IP OpenEdition

This is the first edition of this book. It contains some information previously presented in *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, which supports TCP/IP Version 3 for MVS OpenEdition Applications Feature. This book is new for OS/390 TCP/IP OpenEdition, which provides OpenEdition function for TCP/IP in the OS/390 environment. For information about previously available TCP/IP function, continue to use the TCP/IP Version 3 Release 2 for MVS library.

This book supports OS/390 TCP/IP OpenEdition and the OS/390 family of products and includes messages for the following new components:

- Configuration
- Initialization
- SNMPD
- onetstat
- oping
- SNMP Subagent
- osnmp
- Interface Layer
- orouted
- otracert

FTP-C Server Messages

EZY2632—EZY2800

The following are FTP-C Server messages for OS/390 TCP/IP OpenEdition

Guide for information on the parameters of the FTP.DATA file and FTP C server start options.

Source Data Set: EZAFTPDP

Procedure Name: get_num_val, read_ftpdata, parse_start_options

---

EZY2632E *parameter* **value must be either TRUE or FALSE.**

Explanation: While processing the FTP.DATA file, a parameter in the file was encountered which required a value of either TRUE or FALSE. The actual value specified was something other than TRUE or FALSE. *parameter* is the name of the parameter in error.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for *parameter*. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: get_bool_val

EZY2633E *parameter* **value missing - must be either TRUE or FALSE.**

Explanation: While processing the FTP.DATA file, a parameter in the file was encountered which required a value of either TRUE or FALSE, but which had no value specified. *Parameter* is the name of the parameter in error.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide and to the OS/390 TCP/IP OpenEdition User's Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: get_bool_val

EZY2634E *parameter* **value** *value* **not numeric.**

Explanation: While processing either the FTP.DATA file or processing the start options from the FTP C server start procedure, a parameter was encountered which required a numeric value, but which had a non-numeric value specified. *parameter* is the name of the parameter in error. *value* is the value which was found in the FTP.DATA file or on the start options for the parameter. This message will be preceded by either message EZYFT46E or EZY2655E indicating whether the error was in the start procedure or in the FTP.DATA file.

System Action: The parameter in error is ignored. Processing of the FTP.DATA file continues with the next line in the file. Processing of the FTP C server procedure start options continues with the next start option.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file or FTP C server start procedure corrected.

System Programmer Response: Correct the FTP.DATA file or the FTP C server procedure to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

EZY2635E Integer overflow *number* **for** *parameter*

Explanation: While processing either the FTP.DATA file or the FTP C server procedure start options, a parameter was encountered which required a numeric value, but the value which was specified was larger than the largest valid 4 byte integer value. *parameter* is the name of the parameter in error. *number* is the value which was found in the FTP.DATA file or FTP C server procedure start options for the parameter. This message will be preceded by either message EZYFT46E or EZY2655E indicating whether the error was in the start procedure or in the FTP.DATA file.

System Action: The parameter in error is ignored. Processing of the FTP.DATA file continues with the next line in the file. Processing of the FTP C Server start option continues with the next start option.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file or FTP C server start procedure corrected.

System Programmer Response: Correct the FTP.DATA file or FTP C Server start procedure to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file and FTP C Server start options.

Source Data Set: EZAFTPDP

Procedure Name: get_num_val, read_ftpdata, parse_start_options

---

EZY2636E *parameter* **value not specified.**

Explanation: While processing the FTP.DATA file, a parameter in the file was encountered which required a value, but no value was specified. *parameter* is the name of the parameter in error.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: get_num_val, get_str_val, read_ftpdata, parse_start_options

---

EZY2638I Using FTP configuration defaults.

Explanation: The FTP C Server was unable to locate/open any external FTP.DATA files. The internal defaults for all FTP.DATA file parameters are being used. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

System Action: The configurable parameters of the FTP C Server are set to the server provided default values.

User or Operator Response: None.

System Programmer Response: If values other than the ones provided by the server are desired, create an FTP.DATA file with the desired parameter values. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file. If an FTP.DATA file already exists, the Syslog daemon log file should contain additional FTP Server error messages indicating why the existing FTP.DATA file was not used.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2640I Using *name* **for local site configuration parameters.**

Explanation: Indicates which FTP.DATA file is being used for the configurable server parameters. *name* will either be "DD:SYSFTPD", indicating that the SYSFTP DD statement was used for the FTP.DATA file, or it will be the actual name of the file being used. The search order for the FTP.DATA file is: /etc/ftp.data HFS file, SYSFTP DD statement in the Server start procedure, server_address_space_name.FTP.DATA data set, SYS1.TCPPARMS(FTPDATA) PDS member, tcpip.FTP.DATA data set.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2642E Unknown keyword: *parameter*

Explanation: While processing the FTP.DATA file, the FTP C Server encountered an parameter which it did not recognize.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Check the line in error for spelling or typographical errors. If the parameter is incorrect, correct the parameter as necessary. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2643E *parameter* **value longer than** *number* **characters.**

Explanation: While processing the FTP.DATA file, a parameter in the file was encountered which required a character string no longer than the specified number of characters; However, the value that was specified was longer than the maximum allowed length. *parameter* is the parameter in error. *number* is the maximum allowed length of the parameter value.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata, parse_start_options

EZY2644E *parameter* **value** *number* **out of range. Value must between** *low* **and** *high***.**

Explanation: While processing either the FTP.DATA file or processing the start options from the FTP C Server start procedure, a parameter was encountered which required a numeric value which was within a specified range. The actual value specified was outside of the allowable range of values for the parameter being processed. *parameter* is the parameter in error. *number* is the actual value that was specified. *low* is the minimum acceptable value, *high* is the maximum allowable value. This message will be preceded by either message EZYFT46E or EZY2655E indicating whether the error was in the start procedure or in the FTP.DATA file.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file. Processing of the FTP C Server procedure start parameters continues with the next start parameter.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file or FTP C Server start procedure corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata, parse_start_options

EZY2645E CONDDISP value must be either CATLG or DELETE.

Explanation: While processing the FTP.DATA file, the server encountered the CONDDISP parameter with a parameter value that was not CATLG or DELETE. The only valid values for the CONDDISP parameter are CATLG or DELETE.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2648E FILETYPE value must be either SEQ, JES, or SQL.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the FILETYPE parameter, but the value specified for the FILETYPE parameter was not SEQ, JES, or SQL.

System Action: The line containing the FILETYPE parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the FILETYPE parameter in the FTP.DATA file to be either SEQ, JES, or SQL. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2651E UNITNAME parameter *unit*: invalid unit type.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the UNITNAME parameter, but the value specified for the UNITNAME parameter was not a valid DASD or TAPE unit as defined by the host MVS system.

System Action: The line containing the UNITNAME parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the UNITNAME parameter in the FTP.DATA file to be a valid DASD or TAPE unit. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2654I Extra data has been enabled for general tracing.

Explanation: The MODIFY option, DUMP, has been issued to request additional detailed data be captured in the general trace. If general tracing is currently active, FTP begins immediately to include data such as parameter lists and storage areas in the trace log. If the general trace is not currently active, the last setting of detail level (DUMP or NODUMP) will be honored when the general trace is started.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2655E Error in start parameters.

Explanation: While processing the start options specified in the FTP C Server start procedure, the FTP C Server encountered an error.

System Action: The parameter in error is ignored. Processing of the FTP C Server start options continues with the next start option.

User or Operator Response: This message should be followed by another message which describes the error. Contact the System programmer with both messages to have the FTP C Server start procedure corrected.

System Programmer Response: This message should be followed by another message which describes the error. Correct the error described by the second message.

Source Data Set: EZAFTPDP

Procedure Name: parse_start_options

EZY2656I Extra data has been disabled for general tracing.

Explanation: The MODIFY option, NODUMP, has been issued to request that additional detailed data be excluded from the general trace. If general tracing is currently active, FTP begins immediately to exclude data such as parameter lists and storage areas from the trace data. If the general trace is not currently active, the last setting of detail level (DUMP or NODUMP) will be honored when the general trace is started.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2657W Host name unknown; gethostname() error: *error*

Explanation: The gethostname() socket call did not complete successfully.

System Action: Processing continues. The hostname will not be known to the server.

User or Operator Response: Correct the error indicated by the *error*.

System Programmer Response: Correct the error indicated by the *error*.

Source Data Set: EZAFTPDM

Procedure Name: main

EZY2658W Host domain name unknown: gethostbyname error.

Severity: warning

Explanation: The gethostbyname() socket call did not complete successfully.

System Action: Processing continues. The host domain will not be known to the server.

User or Operator Response: Contact your system programmer.

System Programmer Response: Ensure that the host name is accessible via a name server, or defined in hlq.HOSTS.SITEINFO and hlq.HOSTS.ADDRINFO. Refer to OS/390 TCP/IP OpenEdition Configuration Guide for information on domain name systems or HOSTS.SITEINFO and HOSTS.ADDRINFO data sets.

Source Data Set: EZAFTSD

Procedure Name: main

EZY2659E RECFM value must be one of: F, FA, FB, FBA, FBM, FBS, FM, V, VA, VB, VBA, VBM, VBS, VM$, VS, or U.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the RECFM parameter, but the value specified for the parameter was not a valid value. The value must be one of the values listed in the message.

System Action: The line containing the RECFM parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the RECFM parameter in the FTP.DATA file to be a valid record format. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2660E $ MODIFY parameter *parameter* is not valid while User Trace is active.

Explanation: FTP is currently tracing a specified user ID. The only MODIFY parameters accepted when FTP is tracing a specified user ID are another UTRACE to end tracing one user ID and begin tracing another user ID, or NOUTRACE to end user tracing. The MODIFY command is rejected.

System Action: FTP continues.

User or Operator Response: If you want to use a MODIFY option other than UTRACE or NOUTRACE, you must first issue a MODIFY command for NOUTRACE.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2661E A userid must be specified. MODIFY command ignored.

Explanation: A MODIFY command with the UTRACE option was entered, but a user ID was not specified. The UTRACE option must be followed (with no intervening blanks) by =*userid* where *userid* is the user ID to be traced. The MODIFY command is rejected.

System Action: FTP continues.

User or Operator Response: Reenter the command specifying a user ID.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2662E Userid *userid* has too many characters.

Explanation: A command was entered specifying a user ID with more than 8 characters. The command is rejected.

System Action: FTP continues.

User or Operator Response: Reenter the command specifying a user ID of up to 8 characters.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2664I User trace has been enabled for userid *userid*.

Explanation: A MODIFY command was entered requesting detailed tracing of all FTP activity for the specified user ID. Detailed tracing is begun for the specified user ID. FTP activity for other user IDs will not generate trace entries.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2665I User trace has been disabled for userid *userid*.

Explanation: Tracing for user ID *userid* has stopped. This is due either to a MODIFY jobname,NOUTRACE command which has terminated user tracing, or to a MODIFY jobname,UTRACE=userid command which has switched tracing from one userid to another.

System Action: Tracing for *userid* halts. If a MODIFY jobname,NOUTRACE was issued, tracing options are returned to the settings in effect prior to the MODIFY jobname,UTRACE=*userid* command which initiated user tracing for *userid*.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2666I User trace is not active.

Explanation: A MODIFY command was entered requesting that user tracing be ended, but user trace is not currently active. The MODIFY command is ignored.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2669E SPACETYPE value must be either BLOCK, TRACK, or CYLINDER.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the SPACETYPE parameter, but the value specified for the parameter was not a valid value. The value must be one of the values listed in the message.

System Action: The line containing the SPACETYPE parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the SPACETYPE parameter in the FTP.DATA file to be a valid space type. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2670E SQLCOL value must be either NAMES, LABELS, or ANY.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the SQLCOL parameter, but the value specified for the parameter was not a valid value. The value must be one of the values listed in the message.

System Action: The line containing the SQLCOL parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the SQLCOL parameter in the FTP.DATA file to be a valid column heading. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2671I General tracing is already active.

Explanation: A MODIFY command with the TRACE option has been issued while tracing of FTP's general activity is currently active. General tracing continues.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2672E read_ftpdata invalid keycode encountered: *key_code*

Explanation: While processing the FTP.DATA file, an invalid internal key_code was encountered. The value of the key_code is listed in the message.

System Action: An assertion failure occurs to provide additional information on the error. Processing continues with the next statement in the FTP.DATA file.

User or Operator Response: Contact the IBM System Support Center with the error message and the assertion failure output.

System Programmer Response: Contact the IBM System Support Center with the error message and the assertion failure output.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2673E Error reading FTP configuration file: *error*

Explanation: While reading the records in the FTP.DATA set, the read of one of the records failed with the specified error.

System Action: Processing of the FTP.DATA file is terminated.

User or Operator Response: Correct the problem with the FTP.DATA file.

System Programmer Response: Correct the problem with the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2674E LRECL *number* **is invalid for RECFM** *recfm***.**

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM and LRECL variables and found them to be set to a combination that is invalid for the MVS operating system.

System Action: The LRECL value is changed to the nearest value which is compatible with the record format. This message should be followed by EZY2675I, indicating the new value of the LRECL.

User or Operator Response: If the new value is acceptable, no action is necessary, otherwise contact the system programmer to get the correct lrecl and record format values put into the FTP.DATA file.

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM and / or LRECL. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2675I LRECL being reset to *number*

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM and LRECL parameters and found them to be set to a combination that is invalid for the MVS operating system. The LRECL parameter was changed to the indicated lrecl. This message is usually preceded by message EZY2674E indicating the error.

System Action: LRECL parameter is changed.

User or Operator Response: Correct the error as indicated for message EZY2674E.

System Programmer Response: Correct the error as indicated for message EZY2674E.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2676E BLOCKSIZE must equal LRECL for RECFM *recfm***.**

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM, LRECL, and BLOCKSIZE parameters and found the RECFM parameter set to a record format which requires the LRECL and BLOCKSIZE parameters to be equal, but they were not equal. The BLOCKSIZE parameter was changed to the indicated lrecl. This message is followed by message EZY2677I indicating the new value of BLOCKSIZE.

System Action: The blocksize value is changed to the lrecl value.

User or Operator Response: If the new value is acceptable, no action is necessary, otherwise contact the system programmer to get the correct lrecl, blocksize, and record format values put into the FTP.DATA file.

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM and / or LRECL and / or BLOCKSIZE. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2677I BLOCKSIZE being set to *blocksize number***.**

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM, LRECL, and BLOCKSIZE parameters and found the RECFM parameter set to a record format which requires the value BLOCKSIZE parameter to have a dependency on the value of the LRECL parameter, but the dependency was not met. The BLOCKSIZE parameter was changed to comply with the dependency on the LRECL. This message is preceded by message EZY2676E, EZY2678E, or EZY2679E indicating the nature of the relationship between BLOCKSIZE and LRECL.

System Action: The blocksize value is changed to the indicated value.

User or Operator Response: Correct the error as indicated for message EZY2676E, EZY2678E, or EZY2679E.

System Programmer Response: Correct the error as indicated for message EZY2676E, EZY2678E, or EZY2679E.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2678E BLOCKSIZE must be a multiple of LRECL for RECFM *recfm***.**

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM, LRECL, and BLOCKSIZE parameters and found the RECFM parameter set to a record format which requires the BLOCKSIZE to be a multiple of the LRECL parameter, but it was not.

System Action: BLOCKSIZE is set to a value that is the nearest multiple of LRECL.

User or Operator Response: If the new value is acceptable, no action is necessary, otherwise contact the system programmer to get the correct lrecl, blocksize, and record format values put into the FTP.DATA file.

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM and / or LRECL and / or BLOCKSIZE. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2679E BLOCKSIZE must be at least 4 more than LRECL for RECFM *recfm*

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM, LRECL, and BLOCKSIZE parameters and found the RECFM parameter set to a record format which requires the BLOCKSIZE to be at least 4 greater than the LRECL parameter, but it was not.

System Action: BLOCKSIZE is set to LRECL + 4. If LRECL is greater than 32756, the LRECL will be changed to 32756 so that BLOCKSIZE will not exceed the maximum value of 32760.

User or Operator Response: If the new value is acceptable, no action is necessary, otherwise contact the system programmer to get the correct lrecl, blocksize, and record format values put into the FTP.DATA file.

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM and / or LRECL and / or BLOCKSIZE. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2680I LRECL being changed to *number*

Explanation: After processing the FTP.DATA file, the server cross checked the new values of the RECFM, LRECL, and BLOCKSIZE parameters and found the RECFM parameter set to a record format which requires the BLOCKSIZE to be at least 4 greater than the LRECL parameter, but it was not. The server attempted to change the BLOCKSIZE value to LRECL + 4, but the LRECL value was greater than 32756, which would cause the BLOCKSIZE to exceed the maximum value of 32760. LRECL was changed to the indicated value.

System Action: LRECL is changed to the indicated value.

User or Operator Response: Correct the error as indicated for message EZY2679E.

System Programmer Response: Correct the error as indicated for message EZY2679E.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2681E Invalid record format 'recfm' encountered.

Explanation: After processing the FTP.DATA file, the server attempted to cross check the RECFM, BLOCKSIZE, and LRECL values, but the value of RECFM was invalid.

System Action: The RECFM, BLOCKSIZE, and LRECL values are reset to the default values.

User or Operator Response: Contact the IBM System Support Center with the error message.

System Programmer Response: Contact the IBM System Support Center with the error message.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2682I LRECL, RECFM, and BLOCKSIZE being reset to default values.

Explanation: After processing the FTP.DATA file, the server attempted to cross check the RECFM, BLOCKSIZE, and LRECL values, but the value of RECFM was invalid. The LRECL, RECFM, and BLOCKSIZE values are reset to the default values.

System Action: The RECFM, BLOCKSIZE, and LRECL values are reset to the default values.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZY2683I Extra data has been enabled for JES tracing.

Explanation: The MODIFY option, JDUMP, has been issued to request additional detailed data be captured in the JES trace.

System Action: If JES tracing is currently active, FTP begins immediately to include data such as parameter lists and storage areas in the JES trace log. If the JES trace is not currently active, the last setting of detail level (JDUMP or NOJDUMP) will be honored when the JES trace is started. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2684I Extra data has been disabled for JES tracing.

Explanation: The MODIFY option, NOJDUMP, has been issued to request that additional detailed data be excluded from the JES trace.

System Action: If JES tracing is currently active, FTP begins immediately to exclude data such as parameter lists and storage areas from the JES trace. If the JES trace is not currently active, the last setting of detail level (JDUMP or NOJDUMP) will be honored when the JES trace is started. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2688E Unknown start option: *option*

Explanation: While processing the start options specified in the FTP C Server start procedure, the FTP C Server encountered the invalid start option listed in the message.

System Action: The start option is ignored. Processing continues with the next start option.

User or Operator Response: Contact the system programmer to correct the FTP C Server start procedure.

System Programmer Response: Correct the start option in the FTP C Server start procedure. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the FTP C Server start options.

Source Data Set: EZAFTPDP

Procedure Name: parse_start_options

EZY2689E An error occurred processing the FTP translate table file *name*.

Explanation: FTP encountered an error while attempting to process the translation table file *name*.

System Action: The FTP server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: This message is followed by an additional message that further defines the error. Correct the error indicated by these messages.

System Programmer Response: This message is followed by an additional message that further defines the error. Correct the error indicated by these messages.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZY2690E Header record invalid format.

Explanation: The FTP C Server was attempting to load a translation table, but the file specified for the translation table did not have a valid TCPXLBIN header record. The first record in the file should have been "*TCP/IP translate tables", but was not.

System Action: The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: If the file name is correct, the file has probably been corrupted. Use the CONVXLAT command to rebuild the table.

System Programmer Response: If the file name is correct, the file has probably been corrupted. Use the CONVXLAT command to rebuild the table.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZY2691E Error reading the file: *error*

Explanation: The FTP C Server was attempting to load a translation table, but the error *error* occurred while attempting to read the file.

System Action: The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: Correct the error indicated by *error*. It may be necessary to rebuild the TCPXLBIN file using the CONVXLAT command.

System Programmer Response: Correct the error indicated by *error*. It may be necessary to rebuild the TCPXLBIN file using the CONVXLAT command.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZY2693I Unable to open *name* **:** *reason*

Explanation: The FTP server attempted to open the specified translation table file, but the open failed for the specified reason.

System Action: The FTP server will continue through the search order.

User or Operator Response: If the file *name* is the desired file, correct the error specified by *reason*. This message will be issued for each file as the server proceeds through the search order. If the desired file further in the search order, no action is necessary.

System Programmer Response: If the file *name* is the desired file, correct the error specified by *reason*. This message will be issued for each file as the server proceeds through the search order. If the desired file is further in the search order, no action is necessary.

Source Data Set: EZAFTPDP, EZAFTPDX

Procedure Name: read_ftpdata, setup_translate_tables, read_xlate_files

---

EZY2697I MVS TCP/IP Server-FTP *version time* **on** *date*

Explanation: This is an informational message indicating that the MVS FTP C Server has completed initialization and is ready for client connections. The message indicates the version of the FTP C Server and the time and date the server was started.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZY2700I Using port FTP control *(portnumber)*

Explanation: This is an informational message indicating which port the FTP C Server is listening on for incoming client connections. This should be either the port specified for the ftp server in the etc.services file, the port specified by the PORT start option, or default port 21.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZY2701I Inactivity time is *timeout*

Explanation: This is an informational message indicating the value being used for the client control connection timeout. Client control connections inactive for this number of seconds will be terminated by the server. A timeout value of 0 indicates that inactive client connections will not be terminated. This value should be either the default timeout value or the value specified by the INACTIVE start option or FTP.DATA parameter.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZY2702I Server-FTP: Initialization completed at *time* on *date*

Explanation: This is an informational message indicating that the MVS FTP C Server has completed initialization and is ready for client connections. The message indicates the time and date the server was started.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZY2704I General tracing has been enabled.

Explanation: The command handler has received a Modify request to start tracing. The tracing has been successfully started.

System Action: Trace information will be logged to the Syslog daemon log file specified for daemon.debug.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2705I General tracing has been disabled.

Explanation: The command handler has received a Modify request to stop tracing. The tracing has been disabled.

System Action: No more traces are collected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2706E Unsupported Modify command parameter *parameter* ignored.

Explanation: A Modify request was issued to the Server, but the server did not recognize the parameter entered on the modify request.

System Action: The request is ignored.

User or Operator Response: Verify that the Modify command was entered correctly. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the Modify command.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2707E Unsupported console command (hex code) *command name* ignored.

Explanation: An MVS command was entered to the FTP C Server, but the server did not support the MVS command. Currently the only supported MVS commands are MODIFY and STOP.

System Action: The MVS command is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2710I JES tracing has been enabled.

Explanation: A Modify command was entered to start JES tracing. The JES tracing has been successfully started.

System Action: JES Trace information will be logged to the Syslog daemon log file specified for daemon.debug.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2711I JES tracing has been disabled.

Explanation: A Modify request was received to disable JES tracing. JES tracing has been disabled.

System Action: JES traces are no longer collected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2714E FTP server shutdown in progress

Explanation: The FTP server has been terminated either by an MVS operator STOP command, or by and OMVS kill command.

System Action: The FTP server terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: pgmstpd

EZY2715I General tracing is not active.

Explanation: A MODIFY command was entered requesting that general tracing of all user IDs be ended, but tracing is not currently active. The MODIFY command is ignored.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2716I JES tracing is not active.

Explanation: A MODIFY command was entered requesting that tracing of JES activity for all user IDs be ended, but JES tracing is not currently active. The MODIFY command is ignored.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2720I Using Japanese translation tables in *filename*

Explanation: The translation table in the file is used as specified.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2721E Unable to load Japanese translation tables in *filename*; **return code:** *code*

Explanation: The FTP server encountered an error while attempting to load the indicated translation table. Possible return codes are:

1 No storage for the translation table
2 Unable to find the translation table header
3 Error reading the file

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Respond as indicated by the return code displayed in the message.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2722E Unable to load Japanese translation tables.

Explanation: The FTP server cannot open the indicated translation table.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Make sure that the translation table is properly specified and loaded in storage accessible to the FTP server.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2723I Using Korean translation tables in *filename*

Explanation: The FTP server is using the indicated translation table.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2724E Unable to load Korean translation tables in *filename*; **return code:** *code*

Explanation: The FTP server was unable to load the indicated translation table. Possible return codes are:

1 No storage for the translation table
2 Unable to find the translation table header
3 Error reading the file

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Respond as indicated by the return code displayed in this message.

System Programmer Response: None.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2725E Unable to load Korean translation tables.

Explanation: The FTP server cannot open the indicated translation table.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Make sure that the translation tables are properly specified and in storage accessible to the FTP server.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2726I Using traditional Chinese translation tables in *filename*

Explanation: The FTP server is using the indicated Chinese translation table.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2727E Unable to load traditional Chinese translation tables in *filename*; **return code:** *code*

Explanation: The FTP server cannot open the indicated Chinese translation table. Possible return codes are:

1 No storage for the translation table
2 Unable to find the translation table header
3 Error reading the file

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Respond as indicated by the return code displayed in this message.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2728E Unable to load traditional Chinese translation tables.

Explanation: The FTP server cannot open the indicated Chinese translation table.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Make sure that the Chinese translation table is properly specified and in storage accessible to the FTP server.

Source Data Set: EZAFTPDC

Procedure Name: read_dbcs_table

EZY2729I JES tracing is already active.

Explanation: A MODIFY command with the JTRACE option has been issued while tracing of FTP's JES-related activity is currently active. JES tracing continues.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

Procedure Name: mvs_command_handler

EZY2790E Invalid format data set name *"dsn"*. Dcbdsn parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *dsn*, was not a valid MVS data set name. The DCBDSN keyword is ignored. The FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Correct the DCBDSN data set name in the FTP.DATA file. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2792E Error retrieving *"data_set"*. Dcbdsn parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was migrated, and the server was unable to recall the data set. The keyword is ignored. The FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Determine why the data set cannot be recalled and correct the error.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2793E *"data_set"* is migrated and noautorecall is specified. Dcbdsn parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was migrated and No Autorecall had also been specified. The server was unable to recall the data set. The keyword is ignored. The FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Change the FTP.DATA file or the FTP server start procedure to specify Autorecall; or recall the data set prior to starting the FTP server (note that the data set must then not be migrated at any point while the FTP server is active); or change the data set specified for the DCBDSN keyword to a data set that is not migrated. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2794E Error mounting *"data_set"*. Dcbdsn parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was on an unmounted volume. The server encountered an error attempting to mount the volume. The keyword is ignored and the FTP server continues processing with the next keyword.

System Action: FTP continues.

User or Operator Response: Determine why the volume could not be mounted and correct the error.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2795E Volume for *"data_set"* is not mounted and noautomount is specified. DCBDSN parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was on an unmounted volume and NO AUTOMOUNT had also been specified. The server was unable to mount the volume. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Contact the system programmer to change the settings of the FTP.DATA file. If necessary, mount the volume containing the DCBDSN data set.

System Programmer Response: Change the FTP.DATA file or server job start options to allow Automount; mount the volume containing the DCBDSN data set (note that the volume must then remain mounted for the duration of the FTP server job); or change the DCBDSN parameter to specify a data set that is on a mounted volume. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2796E Data set *"data_set"* does not exist. Dcbdsn parameter ignored.

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, did not exist. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Notify the system programmer of the error.

System Programmer Response: Correct the FTP.DATA file to specify an existing data set for the DCBDSN keyword. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2797E *"data_set"* **is not on a direct access volume. Dcbdsn parameter ignored.**

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was not located on a direct access volume. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Change the DCBDSN parameter of the FTP.DATA file to specify a data set that is on a direct access volume. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2798E *"data_set"* **is a VSAM data set. Dcbdsn parameter ignored.**

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, was a VSAM data set and could not be used as a model data set. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Contact the system programmer to change the settings of the FTP.DATA file.

System Programmer Response: Change the DCBDSN keyword in the FTP.DATA file to specify a non-VSAM data set. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2799E *"data_set"* **invalid dsorg. Dcbdsn parameter ignored.**

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, had an incorrect data set organization and could not be used as a model data set. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: Contact the system programmer to change the settings of the FTP.DATA file.

System Programmer Response: Change the DCBDSN keyword in the FTP.DATA file to specify a data set with a valid data set organization (PS or PO). Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZY2800E Error locating file *"data_set"***. Dcbdsn parameter ignored.**

Explanation: The FTP server was processing the DCBDSN keyword in the FTP.DATA file. The data set name specified for the DCBDSN data set, *data_set*, could not be located. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: If the data set exists, determine why the server was unable to locate it, and correct the error. If necessary, contact the system programmer to change the settings of the FTP.DATA file.

System Programmer Response: Change the DCBDSN keyword in the FTP.DATA file to specify a valid data set. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

EZYFT01—EZYFT52

EZYFT01I Unable to open message catalog *'message catalog'***.** *error text***. Using FTP's default messages.**

Explanation: An attempt was made to open the FTP server's message catalog (ftpdmsg.cat) in the directory determined by the NLSPATH and LANG environment variables, but the catalog could not be opened for the reason explained in *error text*.

System Action: Processing continues. Default messages will be used.

User or Operator Response: None.

System Programmer Response: If a message catalog is desired, stop the server, correct the problem as indicated by the *error text*, and restart the server.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT02E Filename exceeds maximum valid length of *max***.**

Explanation: The FTP server was processing the SBDATACONN statement in the FTP.DATA file. The file name specified is longer than *max* characters. This message is preceded by message EZYFT46E which locates the error in the FTP.DATA file. The keyword is ignored and the FTP server continues with the next keyword.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Correct the file name provided on the SBDATACONN statement in the FTP.DATA file. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT03E The maximum number of REPLYLANGUAGE statements *(reply_max)* have already been processed.

Explanation: The REPLYLANGUAGE keyword has appeared in the FTP.DATA file too many times. *reply_max* is the maximum number of valid REPLYLANGUAGE statements that will be accepted by FTP.

System Action: The REPLYLANGUAGE statement is ignored. Processing continues.

User or Operator Response: None.

System Programmer Response: To avoid this error message, remove excessive REPLYLANGUAGE statements from the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT04E Invalid syntax. The syntax must be: (value1,value2)

Explanation: The value provided for a statement in the FTP.DATA file does not have valid syntax. This message is preceded by message EZYFT46E which locates the error in the FTP.DATA file.

System Action: The statement is ignored. Processing continues.

User or Operator Response: Contact the System programmer with both messages to have the FTP.DATA file corrected.

System Programmer Response: Use the EZYFT46E message to locate the error in the FTP.DATA file, and correct the error described by this message.

System Action: Processing continues.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT05E Unable to open the reply catalog *catalog*: *error*.

Explanation: The FTP server was processing a REPLYLANGUAGE statement in the FTP.DATA file. A reply catalog (ftpdrply.cat) could not be located/opened in the directory indicated by the REPLYLANGUAGE parameter. The corresponding language will not be available for replies. The message is preceded by message EZYFT46E which locates the error in the FTP.DATA file.

System Action: The REPLYLANGUAGE statement is ignored. Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the *error* indicated in the message and restart the server, if the reply catalog is desired.

System Action: Processing continues.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT06I Using internal messages for replies because there is no reply catalog available.

Explanation: FTP was unable to locate or open a reply catalog (ftpdrply.cat). The reply texts contained within the FTP modules will be used for FTP replies.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If an external reply catalog is desired, either place a copy of ftpdrply.cat in the 'C' subdirectory of your nlspath. If you do not have an NLSPATH environment variable set, FTP uses a default of /usr/lib/nls/msg/C.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT07E UMASK value *'value'* not a valid octal value

Explanation: The value, *value*, specified for the UMASK keyword in the ftp.data configuration file was not a valid 3 character octal number. The value of the UMASK keyword must be a 3 character octal number in the range of 000 - 777.

System Action: The keyword is ignored. Processing continues with the next keyword.

User or Operator Response: Notify the system programmer of the problem.

System Programmer Response: Correct the value specified for UMASK in the ftp.data file. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT08W Unable to get port number from etc.services : *error*

Explanation: The FTP server was unable to retrieve its port number from the etc.services file. *error* is the error message returned by the C runtime library for the failing getservbyname() routine.

System Action: Processing continues. The FTP server will attempt to use either the value of the PORT start option, if specified, or the default port of port 21.

User or Operator Response: None.

System Programmer Response: If the PORT start option is specified, or if PORT 21 is an acceptable default port, this message may be ignored. Otherwise, add an entry for the ftp server to the appropriate etc.services file. Refer to the *OS/390 TCP/IP OpenEdition User's Guide* for information on configuring the etc.services file for the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT09I system information for *nodename*: *sysname* version *version* release *release (machine)*

Explanation: This is an informational message describing the MVS host which the ftp server is running on. *nodename* is the name of the node within an implementation-specified communication network. *sysname* is the name of the implementation of the operating system. *version* is the version level of the operating system. *release* is the release level of the operating system. *machine* is the name of the hardware type the system is running on.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT10I C version *version***, release** *release***, modification** *modification***.**

Explanation: This is an informational message containing the version, release, and modification level of the C runtime library being used.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT11I LE version *version***, release** *release***, modification** *modification***.**

Explanation: This is an informational message containing the version, release, and modification level of the LE runtime library being used.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT12E socket error : *error*

Severity: Error.

Explanation: The ftp server encountered an error while attempting to create the socket for the control port. *error* is the C runtime library error message returned for the failing socket() call.

System Action: If TCP/IP is not available (resource temporarily unavailable), the FTP server will try again in 60 seconds. If any other error has occurred, the FTP server is terminated with exit code 0012.

User or Operator Response: None.

System Programmer Response: If *error* indicates that a resource is temporarily unavailable, ensure that TCP/IP has been started. Otherwise, correct the error indicated by *error* and restart the FTP server.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT13E bind error : *error*

Explanation: The ftp server encountered an error while attempting to bind the socket for the control port. *error* is the C runtime library error message returned for the failing bind() call.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPSK

Procedure Name: accept_client.

EZYFT14E listen error : *error*

Explanation: The ftp server encountered an error while attempting to listen on the socket for the control port. *error* is the C runtime library error message returned for the failing listen() call.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT15E selectex error : *error*

Explanation: The ftp server encountered an error while attempting to setup the control port. *error* is the C runtime library error message returned for the failing selectex() call.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT16E accept error : *error*

Explanation: The ftp server encountered an error while attempting to setup the control port. *error* is the C runtime library error message returned for the failing accept() call.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT17E getsockname error : *error*

Explanation: The ftp server encountered an error while attempting to setup the control port. *error* is the C runtime library error message returned for the failing getsockname() call.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT18I Using catalog *'catalog'* **for FTP server messages.**

Explanation: The messages issued by the FTP server will be retrieved from the message catalog in *file*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT19E Unrecognized code page name: *'name'*

Explanation: A CTRLCONN statement in the FTP.DATA file has an invalid value. Valid values include '7bit' or name of a codeset that is recognized as part of the code set converters used by OE. This message follows message EZYFT46E that provides the location of the error.

System Action: The statement is ignored. Processing continues.

User or Operator Response: None.

System Programmer Response: Use EZYFT46E to locate the error within the FTP.DATA file. Refer to *OS/390 C/C++ Programming Guide* for a list of recognized code pages. The code page name must be entered exactly as it appears in the list.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT20E Invalid parameter length. Maximum length for parameter 1 is *max1***. Maximum length for parameter 2 is** *max2***.**

Explanation: An invalid parameter length was encountered while processing a statement in the FTP.DATA file. This message follows message EZYFT46E which provides the location of the error.

System Action: The line in FTP.DATA containing the error is ignored. Processing continues.

User or Operator Response: Contact the System programmer with the error message to have the FTP.DATA file corrected.

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT21I Using catalog *'file'* **for FTP replies.**

Explanation: The FTP replies sent to the client by the FTP server will be retrieved from the catalog in *file*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT22E Unexpected end of file.

Explanation: The FTP C Server was attempting to load a translation table, but an unexpected end of file occurred while attempting to read the file. This message will be preceded by message EZY2689E which will identify the file being read at the time of the error.

System Action: The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: None

System Programmer Response: Determine why end of file was encountered. It may be necessary to rebuild the TCPXLBIN file using the CONVXLATE command.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZYFT23E No conversion available between CTRLCONN parameter *('parameter')* **and the FTP server's code page** *('codepage')***.**

Explanation: The FTP server was processing a CTRLCONN parameter in the FTP.DATA file. The parameter value *(parameter)* does not indicate a valid ASCII choice for establishing translate tables for FTP's control connection. Valid values include '7bit' (specified in lower, upper, or mixed case) and single-byte ASCII code set names that are recognized by the iconv function.

System Action: The CTRLCONN statement is ignored. The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: None.

System Programmer Response: If an iconv-generated translate table is desired for the control connection, correct the CTRLCONN statement and restart the FTP server. Refer to *OS/390 C/C++ Programming Guide* for information on supported code set converters and the code set names recognized by iconv.

(Note: If the FTP server is running in a double-byte code page, the CTRLCONN parameter cannot be used to establish translate tables for the control connection.)

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT24E Unable to set up conversion between *'page1'* **and** *'page2'***.**

Explanation: The FTP server was processing either a CTRLCONN or SBDATACONN parameter in the FTP.DATA file. A code set converter was successfully opened, but an error occurred while attempting to set up single-byte translate tables using the indicated code sets *page1* and *page2*.

System Action: The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the internally provided translation tables are used.

User or Operator Response: None.

System Programmer Response: If the requested code sets are double-byte code sets, the use of CTRLCONN or SBDATACONN to establish translate tables is not supported, and the statement should be removed from the FTP.DATA file. If the code sets are single-byte code sets, an internal error has occurred. Contact the IBM System Support Center with this message and FTP trace output, if it is available.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT25I Using *file* **for FTP translation tables for the control connection.**

Explanation: *file* is the name of the file that was used to set up the translate tables for the control connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: read_xlate_files

EZYFT26I Using 7-bit conversion derived from *'codeset1'* and *'codeset2'* for the control connection.

Explanation: The FTP server has processed a CTRLCONN statement in the FTP.DATA file. The statement indicated a 7-bit table was desired. The codeset *codeset1* was used for the ASCII code set, and *codeset2* was used for the EBCDIC codeset, but only 7-bit translations appear in the translate table.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT27I Using conversion between *'codeset1'* and *'codeset2'* for the control connection.

Explanation: The FTP server has processed a CTRLCONN statement in the FTP.DATA file. The translate table for the control connection was built using codeset *codeset1* for the ASCII code set, and *codeset2* for the EBCDIC codeset.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT28W Unable to use iconv to establish default translate tables for the control connection. Using internal tables.

Explanation: FTP was unable to use iconv to build 7-bit translate tables based on ISO8859-1 and the current host code set. Internal 7-bit tables will be used. This will occur if the FTP server is running in a double-byte code page and no other translate tables were found in the search order.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the FTP server is running in a single-byte code page, an internal error has occurred. Contact the IBM System Support Center with this error message and the output from the FTP server trace.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT29I Using conversion between *'codeset1'* and *'codeset2'* for the data connection.

Explanation: The FTP server has processed a SBDATACONN statement in the FTP.DATA file. The translate table for the data connection was built using codeset *codeset1* for the EBCDIC code set, and *codeset2* for the ASCII codeset.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT30E No conversion available between SBDATACONN parameters: *'parm1'* and *'parm2'*.

Explanation: The FTP server was processing an SBDATACONN parameter in the FTP.DATA file. There is no supported code set converter for the code sets *parm1* and *parm2*. Valid code sets for the first SBDATACONN parameter include the single-byte EBCDIC code set names recognized by the iconv function; valid code sets for the second parameter include the single-byte ASCII code set names.

System Action: The SBDATACONN statement is ignored. The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the translation tables set up for the control connection will also be used for the data connection.

User or Operator Response: None.

System Programmer Response: If an iconv-generated translate table is desired for the data connection, correct the SBDATACONN statement and restart the FTP server. Refer to *OS/390 C/C++ Programming Guide* for information on supported code set converters and the code set names recognized by iconv.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT31I Using *file* for FTP translation tables for the data connection.

Explanation: *file* is either the file name or the ddname for the file that was used to build the translate tables for the data connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables, read_xlate_files, read_xlate_dd

EZYFT32I Using the same translate tables for the control and data connections.

Explanation: The FTP server was not able to set up translate tables following the search order for the data connection: DD: SYSFTSX, SBDATACONN in FTP.DATA, TCPXLBIN file. The same translate table established for the control connection will also be used for the data connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT33I Unable to open DDNAME *'ddname'* for the data connection: *reason*

Explanation: The FTP C Server attempted to open *ddname* for data connection translate tables, but the open failed for the specified reason.

System Action: The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the translate tables set up fro the control connection will also be used for the data connection.

User or Operator Response: If *ddname* is the desired file, correct the error specified by *reason*. If the desired translation table is further in the search order, no action is necessary.

System Programmer Response: If *ddname* is the desired file, correct the error specified by *reason*. If the desired translation table is further in the search order, no action is necessary.

Source Data Set: EZAFTPDX

Procedure Name: read_xlate_dd

EZYFT34W Sigaction for *signal* failed : *error (errcode1/errcode2)*

Explanation: The FTP server encountered an error attempting to setup the signal handler for the signal specified by *signal*. *error* is the error returned by the C runtime library for the failing sigaction() call. *errcode1* and *errcode2* are the C runtime library *errno errno2* values. If the signal handler is not correctly enabled, the server will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

SIGABND handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.

SIGCHLD handler controls cleanup of "zombie" processes when a client connection is terminated. If sigaction fails for SIGCHLD, zombie processes will not be cleaned up when a client connection is terminated.

SIGTERM handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the FTP server will not be cleanly terminated when an MVS operator STOP command is issued or when the server process is "killed".

SIGURG handler controls the processing of Out of Band data, such as the ABOR subcommand. IF sigaction fails for SIGURG, OOB data will not be received or processed by the server.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM, EZAFTPRX, EZAFTPHC

Procedure Name: main, pgmabndc, serve_oob

EZYFT35W NLSPATH environment variable is too long to be used by FTP. The maximum length for FTP's use is *max*. Using FTP's default nlspath: *path*.

Explanation: FTP is unable to use the current setting of the NLSPATH environment variable because it's length is greater than *max*. FTP will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default nlspath *path* is acceptable, ensure that FTP's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so its length is less than *max*, ensure that FTP's catalogs are located in that path, and stop and restart the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: validate_nlspath

EZYFT36W NLSPATH environment variable has more than one instance of '%L'. Using FTP's default nlspath: *path*

Explanation: The NLSPATH environment variable has invalid syntax for FTP's use. Only one instance of '%L' may appear. FTP will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default nlspath *path* is acceptable, ensure that FTP's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so its has at most one instance of '%L', ensure that FTP's catalogs are located in that path, and stop and restart the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: validate_nlspath

EZYFT37W LANG environment variable is longer than *max*. Using FTP's default LANG value: *lang*

Explanation:

Explanation: FTP is unable to use the current setting of the LANG environment variable because it's length is greater than *max*. FTP will use *lang* to locate its message and reply catalogs within the nlspath.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default value *lang* is acceptable, ensure that FTP's catalogs are located there. If it is not acceptable, reset the LANG environment variable so its length is less than *max*, ensure that FTP's catalogs are located in that path, and stop and restart the FTP server.

System Action: Processing continues.

Source Data Set: EZAFTPDM

Procedure Name: validate_lang

EZYFT38W NLSPATH environment variable does not end with '%N'. Using FTP's default nlspath: *path*

Explanation: The NLSPATH environment variable has invalid syntax for FTP's use. '%N' must appear at the end of the string. FTP will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default nlspath *path* is acceptable, ensure that FTP's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so it has a '%N' at the end of the string, ensure that FTP's catalogs are located in that path, and stop and restart the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: validate_nlspath

EZYFT39W NLSPATH environment variable has '%N' that is not at the end of the string. Using FTP's default nlspath: *path*

Explanation: The NLSPATH environment variable has invalid syntax for FTP's use. '%N' must appear at the end of the string. FTP will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default nlspath *path* is acceptable, ensure that FTP's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so 'N' appears only at the end of the string, ensure that FTP's catalogs are located in that path, and stop and restart the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: validate_nlspath

EZYFT40E FTP server initialization failed - *error*

Explanation: The FTP server was unable to successfully complete initialization. *error* is the error returned by the C Runtime Library for the failing function.

System Action: The FTP server is terminated.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *error*

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT41I Server-FTP: process id *pid***, server job name** *jobname*

Explanation: This is an information message indicating the process id *(pid)* and the server job name *(jobname)* of the FTP server after initialization has completed. The jobname can be used in an MVS operator STOP command to stop the FTP server, or in an MVS operator MODIFY command to control tracing for the ftp server. The process id can be used in an OMVS "kill" command to terminate the server. The process id can also be used to identify trace entries for the FTP server in the SYSLOGD output files.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

EZYFT42E Unable to use file *'file'* **for translate tables for the data connection.**

Explanation: The FTP server was unable to load the translate tables from the file *file* that was specified on an SBDATACONN statement in the FTP.DATA file. If FTP tracing was enabled during initialization, additional messages will precede this one with more specific detail about the error encountered. Possible errors include an invalid header record or incorrect file length. (The file must contain exactly 768 bytes of data.)

System Action: The SBDATACONN statement is ignored. The FTP C Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, the translation tables set up for the control connection will also be used for the data connection.

User or Operator Response: None.

System Programmer Response: If file specified on the SBDATACONN statement is desired, determine why it was not usable, correct the problem, and stop and restart the FTP server.

Source Data Set: EZAFTPDX

Procedure Name: setup_translate_tables

EZYFT43E Unrecognized LOADDBCSTABLES parameter: *parameter***. Parameter ignored.**

Explanation: The LOADDBCSTABLES statement in the TCPIP.DATA file contains an invalid parameter *(parameter)*.

System Action: The parameter is ignored by FTP. The rest of the parameters on the statement are processed. Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the LOADDBCSTABLES statement in the TCPIP.DATA file. If the needed keywords were not already present in the LOADDBCSTABLES statement, stop and restart the FTP server.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT44E Translate table is too small.

Explanation: The FTP server encountered an error while processing a TCPXLBIN file. The file must be exactly 768 bytes in length.

System Action: The FTP server continues through the translate table search order, attempting to find a valid translation table.

User or Operator Response: None.

System Programmer Response: Verify that the TCPXLBIN file being used has the correct format.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZYFT45E Translate table is too large.

Explanation: The FTP server encountered an error while processing a TCPXLBIN file. The file must be exactly 768 bytes in length.

This error will occur if a valid translate table is copied and line control characters are added for each line of the table. (That is, the size of the table is now 771 bytes).

System Action: The FTP server continues through the translate table search order, attempting to find a valid translation table.

User or Operator Response: None.

System Programmer Response: Verify that the TCPXLBIN file being used has the correct format.

Source Data Set: EZAFTPDY

Procedure Name: process_xlate_table

EZYFT46E Error in *name* **file: line** *line_number* **near column** *column_number*

Explanation: An error was detected in the FTP.DATA file. *name* is the name of the file being used as the FTP.DATA file. This will be either "DD:SYSFTPD", indicating that the FTP.DATA file is the one specified by the SYSFTPD DD statement, or it will be the actual file name if the FTP.DATA file was not the one specified on the SYSFTPD DD statement. *line_number* is the number of the line in the FTP.DATA file which contains the error. *column_number* is the approximate location of the error within the line.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line in the file.

User or Operator Response: This message should be followed by another message which describes the error. Contact the System programmer with both messages to have the FTP.DATA file corrected.

System Programmer Response: This message should be followed by another message which describes the error. Correct the error described by the second message.

Source Data Set: EZAFTPDP

Procedure Name: get_bool_val, get_num_val, get_str_val, read_ftpdata

---

EZYFT47I *ftp_data* **file, line** *line_number***: Ignoring keyword** *"keyword"***.**

Explanation: While processing the FTP.DATA file, the FTP server encountered a keyword that was valid for another FTP server or the FTP client, but that is unsupported by this server.

System Action: The FTP server ignores the keyword.

User or Operator Response: None.

System Programmer Response: If the FTP.DATA file is used only by the FTP C server for OpenEdition MVS, and is not shared with another server or client that needs the keyword, remove the keyword from the FTP.DATA file. Refer to the OS/390 TCP/IP OpenEdition Configuration Guide for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT48E Error opening STDOUT or STDERR to /dev/null: *error*

Explanation: During FTP server initialization, the FTP server attempted to open STDOUT and STDERR to /dev/null. The open was unsuccessful. *error* is the error message returned by the C runtime library.

System Action: The FTP server continues; however, without the STDOUT or STDIN file, the LIST and NLST commands will be rejected for HFS files.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT49W Unable to open the reply catalog *catalog* **for the** *language* **replies.** *reason*

Explanation: The FTP server was processing a REPLYLANGUAGE statement in the FTP.DATA file, but the *catalog* could not be opened.

System Action: The FTP server continues.

User or Operator Response: None.

System Programmer Response: If the indicated catalog was intended to be used as the default reply catalog, correct the problem indicated by *reason* and restart the FTP server. If the catalog is intended to be used as an alternate reply catalog, ensure that the translated version of ftpdrply.cat is stored in *catalog* when you want it to be available for selection by a SITE LANG subcommand.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT50E JESRECFM value must be one of: F, V, or *.

Explanation: While processing the FTP.DATA file, the FTP C Server encountered the JESRECFM parameter, but the value specified for the parameter was not a valid value. The value must be one of the values listed in the message.

System Action: The line containing the JESRECFM parameter is ignored. Processing continues with the next line in the file.

User or Operator Response: Contact the system programmer with the error message.

System Programmer Response: Correct the value of the JESRECFM parameter in the FTP.DATA file to be a valid record format. Refer to the TCP/IP for MVS Customization and Administration Guide, Configuring the FTP C Server chapter, and to the TCP/IP V3 for OpenEdition MVS Applications Feature Guide, FTP chapter, for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPDP

Procedure Name: read_ftpdata

EZYFT51I OS/390 version *version***, release** *release***, modification** *modification***.**

Explanation: This is an informational message containing the version, release, and modification level of the OS/390 system.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT52I The runtime version was not recognized.

Severity: Informational.

Explanation: This is an informational message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

Procedure Name: main

EZYFT53E Unable to execute FTP server load module *module* **:** *reason*

Explanation: The FTP server issued an execv() for the load module named *module* but the execv() failed for the specified reason. The FTP server is unable to process incoming connections.

System Action: The FTP session for the connecting client is terminated. The FTP deamon remains active, awaiting client connections.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error indicated by *reason*.

Source Data Set: EZAFTPSK

Procedure Name: setup_new_pgm

OE REXEC Messages

EZYRC01—EZYRC13

The following are REXEC messages for OS/390 TCP/IP OpenEdition.

EZYRC01I Calling function orexec with the following:
EZYRC02I Host *AAA*, **user** *BBB*, **cmd** *CCC*, **port** *DDD*

Explanation: This message pair lists the host, user, command, and port that are passed to the REXEC function provided by OE. *AAA* is the host that the string is being passed to, *BBB* is the logon ID of the user on the remote system, *CCC* is the command that is being passed to the remote system, and *DDD* is the port number.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC03E The call to orexec procedure failed.

Explanation: The call to the REXEC procedure that passes the information to the remote host has failed.

System Action: OREXEC ends.

User or Operator Response: Verify the command was typed correctly, or reissue the command with the -d option. If the problem recurs, ensure that INETD is running on the remote host.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC04I Usage: orexec -V -d -l *user* **-p** *pwd*
EZYRC05I options: -
EZYRC06I -? display this message
EZYRC07I -d turn on debug tracing
EZYRC08I -l *usr* **specifies remote login id**
EZYRC09I -p *pwd* **specifies remote password**
EZYRC10I -s *port* **specifies server port**
EZYRC14I -s *port* **fhost command**
EZYRC15I -C uppercase messages
EZYRC11I -V display APAR level
EZYRC12I Example: rexec -d -l guest -p guest hostname ls -l

Explanation: This group of messages is issued when rexec -? has been entered, or an invalid number of options has been entered.

System Action: OREXEC ends.

User or Operator Response: Correct the invalid option and retry the command.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC13I *XXX YYY*

Explanation: *XXX* is the program called (MVS OE REXEC) and *YYY* is the APAR number, or, if there have been no APARs applied, it is base.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

OE REXECD Messages

EZYRD01—EZYRC32

The following are REXECD messages for OS/390 TCP/IP OpenEdition.

EZYRD01W Invalid option in /etc/inet.conf

Explanation: An invalid option has been used in the /etc/inetd.conf file. The allowable options are: -d, -l, -V.
Note: If an invalid option has been specified, none of the valid options will be in effect.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration options in /etc/inetd.conf for the entry exec under the service name column.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD02E Getpeername failure.

Explanation: Getpeername could not return the name of the peer that is connected to the socket.

System Action: REXECD ends.

User or Operator Response: Retry the command later.

System Programmer Response: The name of the peer that is connected to the socket could not be found for one of the following reasons:

1. The argument is not a valid descriptor.
2. The argument is a file, not a socket.
3. The socket is not connected.
4. Insufficient resources were available in the system to perform the operation.
5. The name parameter pointed to memory not in a valid part of process address space.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD03I Remote address = *XX.XX.XX.XX*

Explanation: *XX.XX.XX.XX* is the internet address (in dotted decimal notation) that the login ID connected from. This message is written to the file specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD04E Binary one not received.

Explanation: The first byte of the string that is passed to the server must be a binary 1.

System Action: REXECD ends.

User or Operator Response: Retry the command.

System Programmer Response: The string that is being passed from the client is not in the proper format. Check the string that is being passed from the client.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD05I clisecport = *n*

Explanation: *n* is the number of the secondary port that is being passed from the client. This is the port that REXECD will send the output to. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD06E Unable to create secondary port.

Explanation: The socket subroutine was unable to create a socket in the specified address family and of the specified type.

System Action: REXECD ends.

User or Operator Response: Retry the command.

System Programmer Response: The socket could not be created for one of the following reasons:

1. The addresses in the specified address family cannot be used with this socket.
2. The socket in the specified address family is not supported.
3. The per-process descriptor table is full.
4. Insufficient resources were available in the system to complete the call.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD07E Cannot make second port *port number*

Explanation: The REXECD server was unable to connect to the specified port.

System Action: REXECD ends.

User or Operator Response: Retry the command.

System Programmer Response: The connect socket call to connect with the requested socket on the client failed for one of the following reasons:

1. The Socket parameter is not valid.
2. The Socket parameter refers to a file, not a socket.
3. The specified address is not available from the local machine.
4. The addresses in the specified address family cannot be used with this socket.
5. The socket is already connected.

6. The establishment of a connection timed out before a connection was made.
7. The attempt to connect was rejected.
8. No route to the network or host is present.
9. The specified address is already in use.
10. The Address parameter is not in a writable part of the user address space.
11. The socket is marked as nonblocking. The connection cannot be immediately completed. The application program can select the socket for writing during the connection process.
12. The specified path name contains a character with the high-order bit set.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD08I User is: *XXX*

Explanation: *XXX* is the login ID that has been passed from the remote host. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD09I Command is: *XXX*

Explanation: *XXX* is the command that has been passed from the remote host. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD10E *XXX***: unknown login. cmd =** *YYY*

Explanation: *XXX* is the logon ID that was passed from the remote host and *YYY* is the command that was passed. The login ID was not in the passwd structure. This message is written to the file that is specified by syslog.conf for information and authorization messages, provided information and/or authorization is in the syslogd.conf file.

System Action: REXECD ends.

User or Operator Response: Correct the login ID and retry. If the login ID is correct, contact the System Programmer.

System Programmer Response: If the login ID is correct, it needs to be specified in the password structure.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD11E Login incorrect.

Explanation: The login was incorrect. Either the user ID or password is incorrect.

System Action: REXECD ends.

User or Operator Response: Retry the command.

System Programmer Response: Note: It is considered to be a security violation to tell the client if the user ID or password is incorrect. This generic message indicates that one of them is incorrect.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD12I Name is: *XXX***, user is** *YYY*

Explanation: This is a debug message. *XXX* indicates the login ID has been passed from the client and *YYY* indicates the login ID obtained from the passwd structure. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD13I dir is *XXX*

Explanation: *XXX* is the home directory obtained from the passwd structure. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD14I uid is: *X***, gid is** *Y*

Explanation: *X* is the uid that is obtained from the passwd structure and *Y* is the gid that is obtained from the passwd structure. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for REXECD and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD15E *XX* **wrong password, cmd =** *YY*

Explanation: *XX* is the login ID passed from the client and *YY* is the command that is being attempted. The password that has been passed is incorrect. This message is written to the file that is specified by syslog.conf for error messages, provided error is set up to write out in syslogd. (This message will not be returned to the client).

System Action: REXECD ends.

User or Operator Response: Correct the password and try again.

System Programmer Response: Note: It is considered to be a security violation to tell the client if the user ID or password is incorrect. This message is written to the file specified by syslog.conf and the message EZYRD11E is returned to the user.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD17E *XX* **no home directory. cmd =** *YY*

Explanation: *XX* indicates the login ID and YY indicates the command that is being attempted.

System Action: REXECD ends.

User or Operator Response: Contact your system programmer.

System Programmer Response: No home directory has been set up in the passwd structure for the login ID that is identified in the message. Add a home directory in the passwd structure for this login ID.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD18E No remote directory.

Explanation: No home directory has been set up on the remote host for the logon ID.

System Action: REXECD ends.

User or Operator Response: Contact the system programmer at the remote site and have a home directory set up in the passwd structure.

System Programmer Response: Set up a home directory in the passwd structure.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD19E Cannot make pipe1.

Explanation: Trying to make a pipe to go between the parent and the child process. This pipe could not be made.

System Action: REXECD ends.

User or Operator Response: Retry the failing command and if it still fails contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. The FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD20E Cannot make pipe2.

Explanation: Trying to make a pipe to go between the parent and the child process for the purpose of doing ASCII to EBCDIC conversion. This pipe could not be made.

System Action: REXECD ends.

User or Operator Response: Retry the failing command and if it still fails contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. The FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD21E Cannot make pipe3.

Explanation: Trying to make a pipe to go between the parent and the child process for the purpose of doing ASCII to EBCDIC conversion. This pipe could not be made.

System Action: REXECD ends.

User or Operator Response: Retry the failing command and if it still fails contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD22E Cannot fork; try again.

Explanation: The parent process is trying to fork off a child process. It has not been able to do this.

System Action: REXECD ends.

User or Operator Response: Retry the failing command and if it still fails contact the system programmer.

System Programmer Response: The fork could not be done for one of the following reasons:

1. The total number of processes executing system-wide or by a single user would be exceeded, or the system does not have the resources necessary to create another process.
2. There is not enough space for this process.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD23E error on read socket 0

Explanation: An attempt was made to read data on the stdin socket. This attempt was not successful.

System Action: REXECD ends.

User or Operator Response: Retry the failing command.

System Programmer Response: A read from socket 0 has failed. It has failed to read for one of the following reasons:

1. The FileDescriptor parameter is not a valid file descriptor open for reading.
2. The file was marked for non-blocking I/O, and no data was ready to be read.
3. A read was interrupted by a signal before any data arrived, and the signal handler was installed with an indication that subroutines are not to be restarted.
4. An I/O error occurred while reading from the file system.
5. The process is a member of a background process attempting to read from its controlling terminal, and either the process is ignoring or blocking the SIGTTIN signal or the process group has no parent process.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD24I *XX* **cmd =** *YY*

Explanation: If the -L option has been specified and if the syslog.conf file has been set up to send information or authorization messages to a file then this message will appear in the file.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD25E setgid failed

Explanation: A set group ID subroutine failed.

System Action: REXECD ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The set group ID can fail for one of two reasons:

1. The process does not have appropriate privileges to set the GID.
2. The value of the GID parameter is incorrect.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD26E initgroups failed

Explanation: An initgroups subroutine failed.

System Action: REXECD ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The subroutine can fail for the following reasons:

1. The number of supplementary groups for the specified user plus the basegid group exceeds the maximum number of groups allowed, or an invalid user is specified.
2. An MVS environmental or internal error occurred.
3. The System authorization facility (SAF) had an error.
4. The caller is not authorized, only authorized users are allowed to alter the supplementary group IDs list.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD27E setuid failed

Explanation: A setuid failed.

System Action: REXECD ends.

User or Operator Response: Contact the systems programmer.

System Programmer Response: A setuid failed. The following are the reasons that a setuid can fail:

1. The process is currently not able to change UIDs.
2. The value of uid is incorrect.
3. The process does not have appropriate privileges to set the UID to uid.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD28E Error from execl()

Explanation: An error has occurred in trying to open the shell process to do the requested command.

System Action: REXECD ends.

User or Operator Response: Retry the command. If the problem persists contact the systems programmer.

System Programmer Response: An execl can fail for the following reasons:

1. The new process' combined argument list and environment list has more bytes than the system defined limit ARG_MAX.
2. The process did not have appropriate permissions to run the specified file for one of the following reasons:
   a. The process did not have permission to search a directory named in your path.
   b. The process did not have execute permission for the file to be run.
   c. The file to be run was not a regular file and the system cannot run files of its type.
3. The new process image file has the appropriate permission and has a recognized format, but the system does not support execution of a file with this format.
4. A loop exists in symbolic links.
5. One or more pathname components in path or file does not exist.
6. The new process image file has the appropriate access permission but is not in the proper format.
7. The new process requires more memory than is permitted by the operating system.
8. A directory component of path or file is not really a directory.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD29E *XX* **too long**

Explanation: A string that is being converted from ASCII to EBCDIC is too long.

System Action: REXECD ends.

User or Operator Response: Retry the command and if the problem persists contact the systems programmer.

System Programmer Response: A string that is outside the permissible limits is being passed to REXECD. Check the parameters being passed and if everything looks correct, contact the IBM System Support Center.

Source Data Set: rexecd.c

Procedure Name: getstr()

EZYRD30I usage: rexecd *-dCLV*

Explanation: This message follows EZYRD01W. It displays the valid options that are permitted in /etc/inetd.conf for the rexecd server.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: Correct the options specified in /etc/inetd.conf and from a superuser ID do a kill -1 *pid number* to cause inetd to reread the inetd.conf file.

Source Data Set: rexecd.c

Procedure Name: usage()

EZYRD31I *XXX YYY*

Explanation: *XXX* is the program called (MVS OE REXECD) and *YYY* is the APAR number or, if there have been no APARs applied, it is base.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD32E Cannot make second port

Explanation: The REXECD server was unable to connect to the specified port.

System Action: REXECD ends.

User or Operator Response: Retry the command.

System Programmer Response: The connect socket call to connect with the requested socket on the client failed for one of the following reasons:

1. The Socket parameter is not valid.
2. The Socket parameter refers to a file, not a socket.
3. The specified address is not available from the local machine.
4. The addresses in the specified address family cannot be used with this socket.
5. The socket is already connected.
6. The establishment of a connection timed out before a connection was made.
7. The attempt to connect was rejected.
8. No route to the network or host is present.
9. The specified address is already in use.
10. The Address parameter is not in a writable part of the user address space.
11. The socket is marked as nonblocking. The connection cannot be immediately completed. The application program can select the socket for writing during the connection process.
12. The specified path name contains a character with the high-order bit set.

Source Data Set: rexecd.c

Procedure Name: doit()

OE RPC Messages

EZA4331—EZA4335 and EZYRP35—EZYRP72

The following are RPC messages for OS/390 TCP/IP OpenEdition.

EZA4331E Usage: orpcinfo -n portnum u host prognum versnum
EZA4332I orpcinfo -n portnum t host prognum versnum
EZA4333I orpcinfo p host
EZA4334I orpcinfo -b prognum versnum
EZYRP35I orpcinfo -d prognum versnum

Explanation: The command line syntax is displayed to indicate that the user invoked ORPCINFO with incorrect arguments.

System Action: ORPCINFO exits.

User or Operator Response: Reenter ORPCINFO with the correct control parameter syntax.

System Programmer Response: None.

Source Data Set: ORPCINFO

Procedure Name: Usage()

EZA4335E Orpcinfo: *service* **is unknown service**

Explanation: The program number specified on the command line does not correspond to a known service.

System Action: ORPCINFO exits.

User or Operator Response: Correct the program number. Check that the server is running and that it registers the correct number. Rerun.

System Programmer Response: None.

Source Data Set: orpcinfo

Procedure Name: Getprognum(), getrpcbyname()

EZYRP36E Sorry. You are not root

Explanation: You must have root authority to use the -d option.

System Action: The registration is not deleted.

User or Operator Response: None.

System Programmer Response: Inform the system administrator of the problem.

Source Data Set: orpcinfo

Procedure Name: stderr

EZYRP37E orpcinfo: Could not delete registration for prog *program* **version** *version*

Explanation: orpcinfo failed to receive a positive response from the portmapper to its request to delete the specified program. See accompanying message for reason.

System Action: The registration is not deleted.

User or Operator Response: None.

System Programmer Response: Use other RPC options as appropriate. If necessary, inform the system administrator of the problem.

Source Data Set: orpcinfo

Procedure Name: stderr

EZYRP52E oportmap CALLIT: cannot fork

Explanation: The portmapper was not able to fork as required to process a broadcast request.

System Action: The portmapper ignores the broadcast request and continues.

User or Operator Response: None.

System Programmer Response: Inform the system administrator of the problem.

Source Data Set: oportmap

Procedure Name: stderr

EZYRP66E enablecache: cache already enabled The rpc udp cache has already been enabled.

Explanation:

System Action: The server continues.

User or Operator Response:

System Programmer Response: Revise the program without the redundant call to svcudp_enablecache().

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP67E enablecache: could not allocate cache

Explanation: The server was not able to allocate storge for its internal tables.

System Action: The server continues without the benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Revise the program to fit in the available storage.

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP68E enablecache: could not allocate cache data

Explanation: The server was not able to allocate storage for its internal tables.

System Action: The server continues without the benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Revise the program to fit in the available storage.

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP69E enablecache: could not allocate cache fifo

Explanation: The server was not able to allocate storage for its internal tables.

System Action: The server continues without benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Revise the program to fit in the available storage.

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP70E cache_set: victim not found

Explanation: The server was not able to find a cache reply that its internal tables indicated was there.

System Action: The server continues without the benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Inform the system administrator of the problem.

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP71E cache_set: victim alloc failed

Explanation: The server was not able to allocate storage for its internal tables.

System Action: The server continues without the benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Revise the program to fit in the available storage.

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP72E cache_set: could not allocate new rpc_buffer

Explanation: The server was not able to allocate storage for its cached reply.

System Action: The server continues without the benefit of the cache request.

User or Operator Response: None.

System Programmer Response: Revise the program to fit in the available storage.

Source Data Set: svc_upd.o

Procedure Name: stderr

OE RSHD Messages

EZYRS01—EZYRS46

The following are RSHD messages for OS/390 TCP/IP OpenEdition.

EZYRS01I *XXX YYY*

Explanation: *XXX* is the program called (MVS OE RSHD) and *YYY* is the APAR number or, if there have been no APARs applied, it is base.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS02W Invalid option in /etc/inet.conf

Explanation: An invalid option has been used in the /etc/inetd.conf file. The allowable options are: -d, -l, and -V.
Note: If an invalid option has been specified, none of the valid options will be in effect.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration options in /etc/inetd.conf for the entry shell under the column service name.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS03E Getpeername failure.

Explanation: Getpeername could not return the name of the peer that is connected to the socket.

System Action: RSHD ends.

User or Operator Response: Retry the command later.

System Programmer Response: The name of the peer that is connected to the socket could not be found for one of the following reasons:

1. The argument is not a valid descriptor.
2. The argument is a file, not a socket.
3. The socket is not connected.
4. Insufficient resources were available in the system to perform the operation.
5. The name parameter pointed to memory not in a valid part of process address space.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS04W Setsockopt (SO_KEEPALIVE) failed

Explanation: There was a failure in setting the socket keepalive option in setsockopt.

System Action: RSHD continues.

User or Operator Response: Retry the command.

System Programmer Response: There was an error in setting the socket keepalive option when using setsockopt. This error should not occur. Contact the IBM System Support Center.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS05W Setsockopt (SO_LINGER) failed

Explanation: There was a failure in setting the socket linger option in setsockopt.

System Action: RSHD continues.

User or Operator Response: Retry the command.

System Programmer Response: There was an error in setting the socket linger option when using setsockopt. This error should not occur. Contact the IBM System Support Center.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS06E Malformed from address.

Explanation: The IP address that was received cannot be put in the correct network byte order.

System Action: RSHD ends.

User or Operator Response: Correct the IP address being sent in from the client.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS07W Connection received using IP options (ignored)

Explanation: An attempt was made to put special IP options in the data stream. All special options are ignored.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS08E Setsockopt IP_OPTIONS NULL.

Explanation: An unsuccessful attempt was made to zero out the IP options.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: This error should not occur. Contact the IBM System Support Center.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS09E Connection from *XX* on illegal port *YY*

Explanation: An attempt was made to connect to RSHD using a non-reserved port. The port numbers that are considered to be reserved are 0-1023.

System Action: RSHD ends.

User or Operator Response: The client should attempt the connection on a reserved port.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS10E Connection on illegal port.

Explanation: An attempt was made to connect to RSHD using a non-reserved port. The port numbers that are considered to be reserved are 0-1023.

System Action: RSHD ends.

User or Operator Response: The client should attempt the connection on a reserved port.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS11E Binary one not received.

Explanation: The first byte of the string that is passed to the server is supposed to be a binary 1. This did not happen in this case.

System Action: RSHD ends.

User or Operator Response: Retry the command.

System Programmer Response: The string that is being passed from the client is not in the proper format. Check the first byte of the string from the client for a binary 1.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS12I clisecport = *n*

Explanation: *n* is the number of the secondary port that is being passed in from the client. This is the port that rshd will send the output to. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS13E Cannot get stderr port

Explanation: The server has attempted to open a reserved port to be used for standard error.

System Action: RSHD ends.

User or Operator Response: Try again later.

System Programmer Response: Do a netstat conn to determine if all reserved ports are in use.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS16E Second port not reserved

Explanation: The server has attempted to open a port that is not reserved.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The port number that the client has requested that the server use is not a reserved port number. The client must request a reserved port number.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS18E Cannot make second port *port number*

Explanation: The rshd server was unable to connect to the specified port.

System Action: RSHD ends.

User or Operator Response: Retry the command.

System Programmer Response: The connect socket call to connect with the requested socket on the client failed for one of the following reasons:

1. The Socket parameter is not valid.
2. The Socket parameter refers to a file, not a socket.
3. The specified address is not available from the local machine.
4. The addresses in the specified address family cannot be used with this socket.
5. The socket is already connected.
6. The establishment of a connection timed out before a connection was made.
7. The attempt to connect was rejected.
8. No route to the network or host is present.
9. The specified address is already in use.
10. The Address parameter is not in a writable part of the user address space.
11. The socket is marked as nonblocking. The connection cannot be immediately completed. The application program can select the socket for writing during the connection process.
12. The specified path name contains a character with the high-order bit set.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS19E Cannot make second port

Explanation: The rshd server was unable to connect to the specified port.

System Action: RSHD ends.

User or Operator Response: Retry the command.

System Programmer Response: The connect socket call to connect with the requested socket on the client failed for one of the following reasons:

1. The Socket parameter is not valid.
2. The Socket parameter refers to a file, not a socket.
3. The specified address is not available from the local machine.

4. The addresses in the specified address family cannot be used with this socket.
5. The socket is already connected.
6. The establishment of a connection timed out before a connection was made.
7. The attempt to connect was rejected.
8. No route to the network or host is present.
9. The specified address is already in use.
10. The Address parameter is not in a writable part of the user address space.
11. The socket is marked as nonblocking. The connection cannot be immediately completed. The application program can select the socket for writing during the connection process.
12. The specified path name contains a character with the high-order bit set.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS20I Could not look up address for *XXX*

Explanation: A gethostbyname was done of *XXX* and the system was unable to find the specified name. The failure could be because of one of the following:

1. The host specified by the Name parameter was not found
2. The local server did not receive a response from an authoritative server. Try again later.
3. An irrecoverable error.
4. The requested Name is valid but does not have an Internet address at the name server.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshc

Procedure Name: doit()

EZYRS21I Remote user is: *XXX*

Explanation: *XXX* is the login ID of the user on the remote host. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshc

Procedure Name: doit()

EZYRS22I Local user is: *XXX*

Explanation: *XXX* is the login ID that has been passed from the remote host. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshc

Procedure Name: doit()

EZYRS23I Command is: *XXX*

Explanation: *XXX* is the command that has been passed from the remote host. This message is written to the file that is specified by syslog.conf for debug messages, provided debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS24E *WWW@XXX* **as** *YYY***: Unknown login. cmd =** *ZZZ*

Explanation: The remote user *WWW* at the remote site *XXX* logging on as user ID *YYY* does not have a valid logon ID. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a valid logon ID specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS25E Login incorrect.

Explanation: The login was incorrect.

System Action: RSHD ends.

User or Operator Response: Retry the command.

System Programmer Response: Note: It is considered to be a security violation to tell the client that the user ID or password is incorrect. This generic message indicates that one of them is incorrect.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS26E *WWW@XXX* **as** *YYY***: no home directory. cmd =** *ZZZ*

Explanation: The remote user *WWW* at the remote site *XXX* logging on as user ID *YYY* does not have a home directory specified. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a home directory specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS27E No remote directory.

Explanation: No home directory has been set up on the remote host for the logon ID.

System Action: RSHD ends.

User or Operator Response: Contact the system programmer at the remote site and have a home directory set up in the security database.

System Programmer Response: Set up a home directory in the security database.

Source Data Set: rshc

Procedure Name: doit()

EZYRS28E *WWW@XXX* **as** *YYY*: **permission denied. cmd =** *ZZZ*

Explanation: The remote user *WWW* at the remote site *XXX* logging on as user ID *YYY* does not have a valid logon ID or password. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a logon ID or password specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS30E Logins currently disabled.

Explanation: The systems administrator has disabled logins.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: A file in the /etc directory with the name nologin has been created. This disables all logins except superusers.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS31E Cannot make pipe.

Explanation: Trying to make a pipe to go between the parent and the child process. This pipe could not be made.

System Action: RSHD ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. The FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS32E Cannot make pipe1.

Explanation: Trying to make a pipe to go between the parent and the child process. This pipe could not be made.

System Action: RSHD ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. The FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS33E Cannot make pipe2.

Explanation: Trying to make a pipe to go between the parent and the child process. This pipe could not be made.

System Action: RSHD ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The pipe could not be made for one of the following reasons:

1. The FileDescriptor parameter points to a location outside of the allocated address space of the process.
2. Two file descriptors are already open (OPEN_MAX).
3. The system file table is full, or the device containing pipes has no free i-nodes.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS34E Cannot fork; try again.

Explanation: The parent process is trying to fork off a child process. It has not been able to do this.

System Action: RSHD ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The fork could not be done for one of the following reasons:

1. The total number of processes executing system-wide or by a single user would be exceeded, or the system does not have the resources necessary to create another process.
2. There is not enough space for this process.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS35E Cannot fork; try again.

Explanation: The parent process is trying to fork off a child process. It has not been able to do this.

System Action: RSHD ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The fork could not be done for one of the following reasons:

1. The total number of processes executing system-wide or by a single user would be exceeded, or the system does not have the resources necessary to create another process.
2. There is not enough space for this process.

Source Data Set: rshc

Procedure Name: doit()

EZYRS36I *WWW@XXX* **as** *YYY*: **cmd =** *zzz*

Explanation: If the -L option has been specified and if the syslog.conf file has been set up to send information or authorization messages to a file, then this message will appear in the file.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS37E setgid failed

Explanation: A set group ID subroutine failed.

System Action: RSHD ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The set group ID can fail for one of two reasons:

1. The process does not have appropriate privileges to set the GID.
2. The value of the GID parameter is incorrect.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS38E initgroups failed

Explanation: An initgroups subroutine failed.

System Action: RSHD ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The subroutine can fail for the following reasons:

1. The number of supplementary groups for the specified user plus the basegid group exceeds the maximum number of groups allowed, or an invalid user is specified.
2. An MVS environmental or internal error occurred.
3. The System authorization facility (SAF) had an error.
4. The caller is not authorized, only authorized users are allowed to alter the supplementary group IDs list.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS39E setuid failed

Explanation: A setuid failed.

System Action: RSHD ends.

User or Operator Response: Contact the systems programmer.

System Programmer Response: A setuid failed. The following are the reasons that a setuid can fail:

1. The process is currently not able to change UIDs.
2. The value of uid is incorrect.
3. The process does not have appropriate privileges to set the UID to uid.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS40E Error from execl()

Explanation: An error has occurred in trying to open the shell process to do the requested command.

System Action: RSHD ends.

User or Operator Response: Retry the command. If the problem persists contact the systems programmer.

System Programmer Response: An execl can fail for the following reasons:

1. The new process' combined argument list and environment list has more bytes than the system defined limit ARG_MAX.
2. The process did not have appropriate permissions to run the specified file for one of the following reasons:
   a. The process did not have permission to search a directory named in your path.
   b. The process did not have execute permission for the file to be run.
   c. The file to be run was not a regular file and the system cannot run files of its type.
3. The new process image file has the appropriate permission and has a recognized format, but the system does not support execution of a file with this format.
4. A loop exists in symbolic links.
5. One or more pathname components in path or file does not exist.
6. The new process image file has the appropriate access permission but is not in the proper format.
7. The new process requires more memory than is permitted by the operating system.
8. A directory component of path or file is not really a directory.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS41E *XX* **too long**

Explanation: A string that is being converted from ASCII to EBCDIC is too long.

System Action: RSHD ends.

User or Operator Response: Retry the command and if the problem persists contact the systems programmer.

System Programmer Response: A string that is outside the permissible limits is being passed to RSHD. Check the parameters being passed and if everything looks correct, contact the IBM System Support Center.

Source Data Set: rshd.c

Procedure Name: getstr()

EZYRS42W rshd *-adlnCLV*

Explanation: This message follows EZYRS02W. It displays the valid options that are permitted in /etc/inetd.conf for the rshd server.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: Correct the options specified in /etc/inetd.conf and from a superuser ID, do a kill -1 *pid number* to cause inetd to reread the inetd.conf file.

Source Data Set: rshd.c

Procedure Name: getstr()

EZYRS43E *WWW* **as** *YYY***: Unknown login. cmd =** *ZZZ*

Explanation: The remote user *WWW*, logging on as user ID *YYY*, does not have a valid logon ID. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a valid logon ID specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS44E *WWW* **as** *YYY***: no home directory. cmd =** *ZZZ*

Explanation: The remote user *WWW*, logging on as user ID *YYY*, does not have a home directory specified. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a home directory specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS45E *WWW* **as** *YYY***: permission denied. cmd =** *ZZZ*

Explanation: The remote user *WWW*, logging on as user ID *YYY*, does not have a valid logon ID or password. The command *ZZZ* was specified.

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: The user ID *YYY* will need to have a valid logon ID or password specified in the security database.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS46I *WWW* **as** *YYY***: cmd =** *ZZZ*

Explanation: This message is issued to a file when the -L option is specified and the syslog.conf file has been set up to send information or authorization messages to that file. The remote user *WWW*, logging on as user ID *YYY*, has executed command *ZZZ*.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

OE Telnet Server Messages - Executive Routine

EZYTE00—EZYTE85

The following are Telnet Server executive routine messages for OS/390 TCP/IP OpenEdition.

EZYTE00E Getpeername failure.

Explanation: This message is written to the client as well as the syslog file. Getpeername could not return the name of the peer that is connected to the socket. The connection ends.

System Action: Processing ends.

User or Operator Response: Retry the command later.

System Programmer Response: The name of the peer that is connected to the socket could not be found for one of the following reasons:

1. The argument is not a valid descriptor.
2. The argument is a file, not a socket.
3. The socket is not connected.
4. Insufficient resources were available in the system to perform the operation.
5. The name parameter pointed to memory not in a valid part of process address space.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE01E Cannot translate current code page.

Explanation: An invalid return code was received from the routine initxlate, using the default code page ISO8859-1. The connection ends. This could be the result of the following conditions:

1. Invalid return from iconv_open for building either the ASCII EBCDIC translation table OR the EBCDIC to ASCII translation table.
2. Invalid return from iconv after opening the translate table.

System Action: Processing ends.

User or Operator Response: Look for further information regarding the failure in the syslog file.

System Programmer Response: Correct the problem with the translation tables and reissue the job.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE02E Invalid suboption in /etc/inetd.conf. Option =*option*

Explanation: An invalid suboption is specified in the /etc/inetd.conf file for -D option. The valid suboptions are: 'netdata', 'ptydata', 'report', 'options' or 'all'.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the configuration options in /etc/inetd.conf for the entry otelnetd under the column service name.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE03E Invalid option in /etc/inetd.conf. Option =*option*

Explanation: An invalid option has been specified in the /etc/inetd.conf file for telnetd.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the configuration options in /etc/inetd.conf for the entry otelnetd under the column service name.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE04I catgets *errno msg* **rsn =** *errno2*

Explanation: This message identifies the error number associated with the first catgets.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE05I Trace *debug_mode* **Debug** *diagnostics* **keepalive** *keepalives* **kludgelinemode** *kludgelinemode* **hostinfo** *hostinfo* **Registered host** *registerd_host* **linemode** *alwayslinemode*

Explanation: If -D report is specified, the above messages will be issued. They describe the parameter settings after the /etc/inetd.conf for telnet is read and processed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE06W Setsockopt (SO_KEEPALIVE) failed

Explanation: There was a failure in setting the socket keepalive option in setsockopt.

System Action: Processing continues.

User or Operator Response: Retry the command. If the error persists, contact the System Programmer.

System Programmer Response: There was an error in setting the socket keepalive option when using setsockopt. This error should not occur. Contact the IBM System Support Center.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE07E Errors found during processing /etc/inetd.conf. Check the syslog file for more information.

Explanation: This message is written to the client. It is informing the user that the parameters which were specified for telnet server are invalid.

System Action: Processing continues.

User or Operator Response: Check the /etc/inetd.conf for the parameters specified. Check the stderr file to determine which

parameters were identified as invalid. The supported options will be printed in the syslog file. Correct the error and retry.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: usage()

EZYTE08I Usage: The following are supported options for telnetd. The options are case sensitive; you cannot capitalize them. The following are valid values:
-C
-D (options|report|netdata|pytdata|all)
-h
-k
-l
-n
-t
-U
Each of these options is defined below.

C	**forces messages to be issued in upper case.**
D	**determines specific diagnostic tracing.**
h	**disables printing of banner upont login.**
k	**tells user not to initiate kludgelinemode.**
l	**specifies line mode as desired action.**
n	**disables TCP keep-alives.**
t	**additional tracing requests.**
u	**refuses connections which cannot be mapped back to a symbolic name.**

Explanation: This is a list of all supported options for this release of the telnet server. The session ends.

System Action: Processing ends.

User or Operator Response: Correct the invalid option and reissue the job.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: usage()

EZYTE09E Telnet session is terminating.

Explanation: An error was found during processing the telnet server options. The session ends.

System Action: Processing ends.

User or Operator Response: Correct the invalid option and reissue the job.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: usage()

EZYTE10I terminaltypeok: call tgetent (buf, *terminal type***)**

Explanation: This message is issued if -t is specified for tracing. It identifies the terminal type which is being negotiated for this connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: terminaltypeok()

EZYTE11I doit: host_name *host_name*
doit: IP address *ip_addr*
doit: PORT *port*
doit: host *host*

Explanation: The following are variables which are set as a result of the gethostname() and gethostbyaddr().

host_name correlates to the name server name related to the ip address of the client.

ip_addr correlates to the dotted decimal notation of the client.

port correlates to the port that the client is using

host correlates to the MVS system being telnetted to.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE12E *lusername* **Permission denied.**

Explanation: This message is written to the client as well as the syslog file. The security checks for the specified user ID failed.

System Action: Processing continues.

User or Operator Response: Verify that the specified user ID is valid for the system trying to access. Verify that the password is the correct password for this user ID. Retry the access.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE13E setgid failed
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. A set group ID subroutine failed.

System Action: Processing ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The set group ID can fail for one of two reasons:

1. The process does not have appropriate privileges to set the GID.
2. The value of the GID parameter is incorrect.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE14E initgroups failed
errno message **rsn =** *errno2*

Explanation: An initgroups subroutine failed.

System Action: Processing ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The subroutine can fail for the following reasons:

1. The number of supplementary groups for the specified user plus the basegid group exceeds the maximum number of groups allowed, or an invalid user is specified.
2. An MVS environmental or internal error occurred.

3. The System authorization facility (SAF) had an error.
4. The caller is not authorized, only authorized users are allowed to alter the supplementary group IDs list.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE15E setuid failed
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog. A setuid failed.

System Action: Processing ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: A setuid failed. The following are the reasons that a setuid can fail:

1. The process is currently not able to change UIDs.
2. The value of uid is incorrect.
3. The process does not have appropriate privileges to set the UID to uid.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE16I uid is: *X***, gid is** *Y*

Explanation: *X* is the uid that is obtained from the passwd structure and *Y* is the gid that is obtained from the passwd structure. This message is written to the trace file if -t is specified as a telnet server option.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE17I interrupt()

Explanation: This trace message indicates that the interrupt subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: interrupt()

EZYTE18I sendbrk()

Explanation: This trace message indicates that the sendbrk subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: sendbrk()

EZYTE19I sendsusp()

Explanation: This trace message indicates that the sendsusp subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: sendsusp()

EZYTE20I recv_ayt()

Explanation: This trace message indicates that the recv_ayt subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt()

EZYTE21I doeof()

Explanation: This trace message indicates that the doeof subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doeof()

EZYTE22I herald()

Explanation: This trace message indicates that the herald subroutine has been entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE23E herald: read error
errno message **rsn =** *errno2*

Explanation: An attempt was made on /bin/banner to read in the herald. The read failed.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errno2* for additional information.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE24E herald: open error
errno message **rsn =** *errno2*

Explanation: An attempt was made to open /bin/banner in the herald. The open failed.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errno2* for additional information.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE25E herald: malloc error
errno message **rsn =** *errno2*

Explanation: An attempt was made to malloc /bin/banner in the herald. The malloc failed.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errno2* for additional information.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE26E herald: stat error
errno message **rsn =** *errno2*

Explanation: An attempt was made to stat /bin/banner in the herald. The stat failed.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errno2* for additional information.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE27I login:

Explanation: This message indicates a request from the server to the user to enter the user ID. It is written to the client, not the syslog.

System Action: Processing continues.

User or Operator Response: Specify the correct user ID for the system which you are telnetting.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE28I *lusername* **Password:**

Explanation: This message indicates a request from the server to the user to enter the password. It is written to the client, not the syslog.

System Action: Processing continues.

User or Operator Response: Specify the correct password for the user ID previously issued.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE29I Starting new telnet session. catfd = *catfd*

Explanation: This is the first message issued by the server when diagnostics are specified. It also outputs the file descriptor associated with the message catalog. If this value equals -1, then the default messages will be issued, instead of the message catalog entries.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE30E Invalid return code received from expired_pw().

Explanation: This message is written to the client as well as the syslog file. The password entered for the user ID specified has expired. the MVS SAF routine has been called to verify the new password entered. It complained, resulting in the failure.

System Action: Processing ends.

User or Operator Response: Check for more information on the __passwd function in your syslog. Specify a valid new password and retry.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE31E Caller is not a member of BPX.DAEMON facility.

Explanation: This message is written to the client as well as the syslog file. The return from __passwd() function resulted in an *errno* of EPERM. The user is not a member of the BPX.DAEMON security package.

System Action: Processing ends.

User or Operator Response: Contact your system support programmer to enable you for BPX.DAEMON

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE32W You entered an invalid login name or password.

Explanation: This message is written to the client as well as the syslog file. The return from __passwd() function resulted in an *errno* stating that the password/user ID combination is invalid. The user will get 3 attempts to issue the correct password for the user ID specified. After the third invalid attempt, the connection ends.

System Action: Processing continues until after the third attempt, and then processing ends.

User or Operator Response: Ensure you are issuing the correct password/userid and retry.

System Programmer Response: None. Note: It is considered to be a security violation to tell the client if the user ID or password is incorrect. This generic message indicates that one of them is incorrect.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE33W Password expired.

Explanation: This message is written to the client as well as the syslog file. The return from __passwd() function indicates that the password entered has expired.

System Action: Processing continues.

User or Operator Response: Enter a new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE34I Enter new password.

Explanation: The return from __passwd() function indicates that the password entered has expired. A request for a new password is indicated.

User or Operator Response: Enter new password.

System Action: Processing continues.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE35I Reenter new password.

Explanation: This message is written to the client, not the syslog file. The password, as entered, may contain typos.

System Action: Processing continues.

User or Operator Response: Reenter the password again to check for typos on entering the expired password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE36E You entered an invalid password.

Explanation: This message is written to the client as well as the syslog file. A call was made to __passwd() to change the expired password. The new password specified resulted in errors.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errno2* for more information. Enter new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE37E New passwords do not match.

Explanation: This message is written to the client as well as the syslog file. The second password entered does not match the first password entered.

System Action: Processing continues.

User or Operator Response: Reenter the new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE38E Password too long.

Explanation: This message is written to the client as well as the syslog file. The specified password is too long. Only a maximum of 8 characters can be used for the password.

System Action: Processing continues.

User or Operator Response: Specify a correct password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE39W TERMINAL type = DUMB. No full screen applications are supported

Explanation: This message is written to the syslog during diagnostics. The terminal type specified during telnet negotiation was unsupported. Processing continues but the terminal acts as a dumb terminal. Terminal type identified as 3270 will be changed to DUMB terminals. Full screen applications will not work. Character at a time raw mode applications such as vi will not work because they require full screen cursor support.

System Action: Processing continues.

User or Operator Response: Do not issue any full screen applications.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: getterminaltype()

EZYTE40E cleanup: waitpid failed.
errno message **rsn =** *errno2*

Explanation: The waidpid() provides a general interface that needs to wait for certain child processes that require accumulated resource utilization statistics. A -1 was returned from the waitpid() for the child.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: The following conditions could be the result of an waitpid failure.

1. The calling process has no existing unwaited-for child processes.
2. The wait_status points to an illegal address.
3. The call was interrupted by a caught signal or the signal did not have the SA_RESTART flag set.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE41I cleanup: child exit status = *wait_status*

Explanation: The result of the waitpid specified during cleanup is specified.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE44E Fork utmp error.
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. The parent process is trying to fork off a child process. It has not been able to do this.

System Action: Processing ends.

User or Operator Response: Retry the failing command and, if it still fails, contact the system programmer.

System Programmer Response: The fork could not be done for one of the following reasons:

1. The total number of processes executing system-wide or by a single user would be exceeded, or the system does not have the resources necessary to create another process.
2. There is not enough space for this process.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE45E parent:utmp error *wait_status*

Explanation: The parent process is responding to the waitpid issued for the utmp processing. The parent could not clean up the pid entry for the specified userid.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: The /etc/utmpx file has been updated or modified to remove the user ID associated with this pid number. Note the difference and ignore the error message.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE46E EXECL: error
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. An error has occurred in trying to open the shell process to do the requested command.

System Action: Processing ends.

User or Operator Response: Retry the command. If the problem persists, contact the systems programmer.

System Programmer Response: An execl can fail for the following reasons:

1. The new process' combined argument and environment list has more bytes than the system defined limit ARG_MAX.
2. The process did not have appropriate permissions to run the specified file for one of the following reasons:
   a. The process did not have permission to search a directory named in your path.
   b. The process did not have execute permission for the file to be run.
   c. The file to be run was not a regular file and the system cannot run files of its type.
3. The new process image file has the appropriate permission and has a recognized format, but the system does not support execution of a file with this format.
4. A loop exists in symbolic links.
5. One or more pathname components in path or file does not exist.
6. The new process image file has the appropriate access permission but is not in the proper format.
7. The new process requires more memory than is permitted by the operating system.
8. A directory component of path or file is not really a directory.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE47I Int: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was input from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE48I Ont: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was output from the telnet server to the client via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE49I Ipt: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was input from the child to the telnet server via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE51W terminaltypeok: Tgetent failure

Explanation: The terminal type, which was passed to the CURSES routine tgetent, did not match the supported types. Processing will attempt to negotiate a new terminal type. The supported terminal types can be found usr/lib/terminfo/ibm.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: terminaltypeok()

EZYTE52E Couldn't resolve your address into a host name. IP address is *addr*
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. The function call to a gethostbyaddr() resulted in a failure. gethostbyaddr() returns a pointer to an object describing an internet host referenced by address. This structure contains the information obtained from the name server. This message is followed by EZYTE11I which identifies the ip address that could not be found.

System Action: Processing ends if -U is specified, otherwise, processing continues.

User or Operator Response: None.

System Programmer Response: Check the *errno* or the *errno2* for more information. The following are possible reasons for failure:

1. No such host is known.
2. The local server did not receive a response from an authoritative server. Retry at some later time.
3. Some unexpected server failure was encountered.
4. The requested name is valid but does not have an IP address associated with this name.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE53E Gethostname Failure.
errno message **rsn =** *errno2*

Explanation: A failure resulted from the gethostname() function call.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Possible causes of this failure are:

1. The name or namelen paramter gave an invalid address.
2. The caller tried to set the hostname and was not a super user.

Check the *errno* and *errno2* for more information.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE54E All network ports in use.

Explanation: This message is written to the client as well as the syslog file. An attempt was made to identify ttys for the child process. An associated tty could not be obtained.

System Action: Processing ends.

User or Operator Response: Wait until some network ports are available.

System Programmer Response: Ensure that you have enough /dev/ptypXXXX and /dev/ttypXXXX for the number of users accessing your system. Examine the ports in use and ensure that none of them are zombies. Eliminate any zombies if found to free up some more ports.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE55E Getpty failed.

Explanation: An attempt was made to identify ttys for the child process. An associated tty could not be obtained.

System Action: Processing ends.

User or Operator Response: Wait until some network ports are available.

System Programmer Response: Ensure that you have enough /dev/ptypXXXX and /dev/ttypXXXX for the number of users accessing your system. Examine the ports in use and ensure that none of them are zombies. Eliminate any zombies if found to free up some more ports.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE56I Yes

Explanation: This message is written to the client, not the syslog file. A request was issued for AYT (are you there?) by the user.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt()

EZYTE57I TELNETD: netwrite *len* **chars**

Explanation: The software issued data to be written to the client. This identifies the number of characters transmitted over the socket.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt()

EZYTE58E verify_password: Getpwnam failed
errno message **rsn =** *errno2*

Explanation: A getpwnam was issued on the user ID specified. The getpwnam failed. The function getpwnam() search the password database for the given login name always returning the first one encountered.

System Action: Processing ends.

User or Operator Response: Ensure the proper user ID was specified upon login. Look to the *errno* and *errno2* specified for more clues on the reason of failure.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE59I read_pw: Character ignored *char*

Explanation: Telnet was reading in information entered by the user (usually user ID or password). There is an extraneous byte of information received after the carriage return (e.g. new line). This byte is ignored and processing is continued.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTE60I cleanup(): child_pid = *child_pid*

Explanation: The child_pid specified is associated with the spawn issued at the start of this connection for the child.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE61E parent: waitpid failed, status = *wait_status*

Explanation: The waitpid() provides a general interface that needs to wait for certain child processes that require accumulated resource utilization statistics. A -1 was returned from the waitpid() for the child.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: The following conditions could be the result of an waitpid failure.

1. The calling process has no existing unwaited-for child processes.
2. The wait_status points to an illegal address.
3. The call was interrupted by a caught signal or the signal did not have the SA_RESTART flag set.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE62E fcntl failure SETOWN

Explanation: A fcntl was issued on the socket file descriptor to enable telnet to now own that socket. This resulted in a failure. A -1 was returned from the fcntl().

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: The following conditions could be the result of a waitpid failure.

1. The argument SETOWN is invalid.
2. The file descriptor specified is invalid.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTE65I PROTOCOL: send IAC Data Mark.

Explanation: A data mark was sent to the client by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE66I PROTOCOL: lmodetype = *lmodetype***, linemode =** *linemode***, uselinemode =** *uselinemode***,**

Explanation: **lmodetype** signifies whether the client can handle real linemode or if use of kludgelinemode is needed. It will be set to one of the following:

0x04 REAL-LINEMODE - use the linemode option

0x03 KLUDGE-OK

0x02 NO-AUTOKLUDGE

0x01 KLUDGE-LINMEODE - use kludge linemode

0x00 NO-LINMODE - client is ignorant of linemode

linemode is true if linemode is currently on.

uselinemode is the state that we wish to be in.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE67I S(nfd): socketfd..ibits=*ibits* **obits=***obits* **ebits=***ebits*
S(nfd): pty...ibits=*ibits* **obits=***obits* **ebits=***ebits*

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the read, write and exceptions on either the masterfd (pty) or the socketfd. They are associated with what will happen within the protocol function. They determine if reading/writing of the master file descriptor or the socket file descriptor will occur. They will also determine if there is an exception outstanding on a particular pty which needs to be addressed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE68I Ept: #bytes = *cc* **pkcontrol(cntl)** *cntl*

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty. An exception condition was identified which needs to be processed. This identifies the number of bytes read in for the exception and what the actual exception condition was which needs to be handled.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE69I PROTOCOL: cntl |= oobdata *cntl*

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty. The exception condition results after it has been processed with out-of-band data.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTS70W PROTOCOL: Found an unknown exception.

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty. An exception was read which is not currently processing. It is ignored. The ones currently handled are:

FLUSHWRITE

DOSTOP

NOSTOP

CHCP

IOCTL

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE71I SYNCHing is turned on *SYNCHing*

Explanation: This messages is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE72I PROTOCOL: netread *ncc* **chars.**

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It identifies the number of bytes read in from the client to be either sent on to the child or processed by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE73I PROTOCOL: EOF on socket.

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE74W PROTOCOL: EOF on master tty.

Explanation: This message is only issued if -t is specified in /etc/inetd.conf. It correlates to the conditions of the exceptions on the pty. An EOF was received on the master tty; as a result, the cleanup () routine is called and the session ends.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Trace the child process to determine why an unexpected EOF was found.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE78I PROTOCOL: ptyread *pcc* **chars**

Explanation: This message is only issued if -D report or -D netdata specified in /etc/inetd.conf. It correlates to the number of bytes read in from the child. It will be sent on to the client after processing by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE79E PROTOCOL: select mask too small, increase FD_SETSIZE

Explanation: This message is written to the client as well as the syslog file. Select uses bit masks of file descriptors in longs. FD_SETSIZE is defaulted to 2048.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Increase the FD_SETSIZE to an appropriate number and rebuild the server.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE80E PROTOCOL: select
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. An unexpected value was returned from the select.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* for additional information on this error. The following a list of possible errors:

1. One of the descriptor sets specified an invalid descriptor.
2. A signal was delivered before the time limit expired and before any of the selected events occurred.
3. The specified time limit is invalid. One of its components is negative or too large.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE81E PROTOCOL: ioctl net FIONBIO
errno message **rsn**=*errno2*

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Check the *errno* and the *errno2* for additional information on the failure. The following are some possible explanations of the failure:

1. The file descriptor specified is invalid.
2. The file descriptor specified is not associated with a character special device.
3. The specified request does not apply to the kind of object that the socket file descriptor references.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE82E PROTOCOL: error fcntl masterfd FIONBIO
errno message **rsn**=*errno2*

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Check the *errno* and the *errno2* for additional information on the failure. The following are some possible explanations of the failure:

1. The file descriptor specified is invalid.
2. The file descriptor specified is not associated with a character special device.
3. The specified request does not apply to the kind of object that the socket file descriptor references.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE83I PROTOCOL: send from socketfd

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE84I PROTOCOL: SIMULATING receive

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE85I SYNCHing = *SYNCHing*

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

OE Telnet Server Messages - Other Messages

EZYTO03—EZYTO09

The following are miscellaneous other Telnet Server messages for OS/390 TCP/IP OpenEdition.

EZYTO03E Incoming session is not from a registered host.

Explanation: The server has been configured with a -U option which stipulates that only registered hosts will be accepted. The telnet session currently being processed is not registered and will be rejected.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the incoming address is specified in the appropriate name server or etc/host file.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTO04I lusername = *lusername*

Explanation: This message is issued during processing of the user ID/password if diagnostics are turned on. It will print out the user ID that was entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTO05I Initial EBCDIC codepage = *__tccp_tonames* **ASCII codepage** = *__tccp_fromname*

Explanation: If diagnostics are on, this message will print out the initial code page settings.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTO06I Int: *20 bytes hex data 10 char ascii.*

System Action: Processing continues.

Explanation: This is hex/ASCII representation of data which was output from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for banner processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTO07I Ont: *20 bytes hex data 10 char ascii.*

System Action: Processing continues.

Explanation: This is hex/ASCII representation of data which was output to the client from the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for banner processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTO08E Password too long.

System Action: Processing continues.

Explanation: This message is written to the client as well as syslog file. Password specified is too long. Only a maximum of 8 characters can be used for the password.

User or Operator Response: Specify a correct password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTO09I options (*count***)** = *options(count)*

Explanation: This message is written to the syslog file. It identifies the options which have been negotiated before the exec() takes place in the telnet server. The available options are specified in the Internet Official Protocol Standards. The resulting value represents what was negotiated:

Value	***Meaning***
3	The client agrees to perform the function.
12	The server agrees to perform the function.

System Action: Processing continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: telnetd.c

Procedure Name: doit()

OE Telnet Server Messages - State Routine

EZYTS01—EZYTS16

The following are Telnet Server state routine messages for OS/390 TCP/IP OpenEdition.

EZYTS01I STATE:telrcv: receive IAC *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation RECEIVED during the execution of the job.

System Action: Telnet server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: telrcv()

EZYTS02I STATE:telrcv: send IAC *c*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation SENT during the execution of the job.

System Action: Telnet server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: telrcv()

EZYTS03E STATE:telrcv: panic state = *state*

Explanation: An IAC command is found during telnet negotiation and is followed by an unrecognized command option.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Issue traces to determine the attempted telnet control data negotiation. Use the'-D options ' trace to issue from the server.

Source Data Set: state.c

Procedure Name: telrcv()

EZYTS04I STATE:send_do: send *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a request to issue a DO option.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_do()

EZYTS05I STATE:willoption: receive WILL *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, the receipt of WILL request.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption()

EZYTS06I willoption: set to kludge ok

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user that kludgelinemode is operational.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption()

EZYTS07I STATE:send_dont: send DON'T *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a request to issue a DONT option.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_dont()

EZYTS08I STATE:wontoption: receive WON'T *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a response that a WONT was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: wontoption()

EZYTS09I STATE:send_will: send WILL *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a response that a WILL was sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_will()

EZYTS10I STATE:dooption: receive DO *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a response that a DO was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dooption()

EZYTS11I STATE:send_wont: send WON'T *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a response that a WON'T was sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_wont()

EZYTS12I STATE:dontoption: receive DON'T *option*

Explanation: This message is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job; specifically, a response that a DON'T was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dontoption()

EZYTS13I ENVIRON VALUE and VAR are reversed!

Explanation: This message is only issued if the user has specified -D options in the /etc/inetd.conf file. It is issued during suboption negotiation. The VALUE and VAR values cannot be reversed for interoperability to occur. If the first recognized character is a VAR or VALUE, the type of client can be identified. If the first recognized character is a USERVAR, the suboption is scanned for 2 consecutive VAR or VALUE fields. (There should not be 2 consecutive VALUE fields). If a client has sent a well-formed option, the number of VALUEs received should always be less than or equal to the number of VARs and USERVARs received. If that is not the case, then the client has reversed the definitions.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: suboption()

EZYTS14I Opt: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was output from the client to the child via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: telrcv()

EZYTS15I STATE:dooption:deferred receive DO *option*

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job, specifically, a response that a DO was received prior to the creation of the tty which may require special tty processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dooption()

EZYTS16I STATE:willoption:deferred receive WILL *option*

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job, specifically, a response that a WILL was received prior to the creation of the tty which may require special tty processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption()

OE TELNET Server Messages - Utility

EZYTU01—EZYTU32

The following are Telnet Server utility for OS/390 TCP/IP OpenEdition.

EZYTU01E UTILITY:Read from ttloop.
errno message **rsn =** *errno2*

Explanation: The number of bytes returned from a read on the socketfd (client) is invalid. A negative value was received. The connection ends. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: Check the *errno* reason code and the *errno2* reason code and correct the problem.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop()

EZYTU02E UTILITY:Read from ttloop. Peer died.
errno message **rsn =** *errno2*

Explanation: The number of bytes returned from a read on the socketfd (client) is invalid. A value of zero was received. The connection ends. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: Check the *errno* reason code and the *errno2* reason code and correct the problem.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop()

EZYTU03I UTILITY:ttloop read *ncc* **chars.**

Explanation: This message is issued if -D report or -D ptydata is specified. It writes the number of characters which were read in from the socketfd (client's socket) to be processed. It is followed by hex data, preceded by the tag "Int". This signifies a hex and ASCII representation of input data coming into the Telnet server from the net.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop()

EZYTU04E Telnetd: UTILITY: Cannot translate current code page.

Explanation: This message is written to the client as well as the syslog file. An error was found while processing the initxlate() routine. The function will end the connection.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check for additional messages issued.

Source Data Set: utility.c

Procedure Name: ReturnToDefault()

EZYTU05E UTILITY: __tcsetcp.
errno message **rsn =** *errno2*

Explanation: An error was found while processing the default code pages. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* reason code and the *errno2* reason code and correct the problem.

Source Data Set: utility.c

Procedure Name: ReturnToDefault()

EZYTU06E Error using code pages *newNames.__tccp_toname* **and** *newNames.__tccp_fromname* **- returning to default code pages.**

Explanation: This message will be issued to the client and only to syslog file, if debug_mode is specified. The current code pages are not working correctly. Telnet is returning to the default code pages. If -D netdata is specified, this message will be followed by the hex and ASCII translation of this message preceded by the tag 'Ont'.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the installation of the code pages. If the requested option is not valid, generate the appropriate codepage. Otherwise, if the option is valid, check to see if the codepage is still valid.

Source Data Set: utility.c

Procedure Name: ReturnToDefault()

EZYTU07E UTILITY: __tcgetcp.
errno message **rsn =** *errno2*

Explanation: An error was found while changing the code pages. This message is written to the client as well as the syslog file.

System Action: The connection ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* reason code and the *errno2* reason code and correct the problem.

Source Data Set: utility.c

Procedure Name: change_translate()

EZYTU08I Telnetd: UTILITY: Change to binary mode.

Explanation: This message indicates that the tty has been switched to binary mode.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: change_translate()

EZYTU09I UTILITY: Change to SingleByte pages
newNames.__tccp_toname
newNames.__tccp_fromname

Explanation: This is a single byte code page. A call to initxlate() will be made to set up the new code pages.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: change_translate()

EZYTU10I UTILITY: Change to MultiByte pages
newNames.__tccp_toname
newNames.__tccp_fromname

Explanation: This is a multi-byte code page. A call to doMultiByte() will be made to set up the new code pages.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: doMultiByte()

EZYTU11E UTILITY: Unexpected input string
errno message **rsn =** *errno2*

Explanation: The *errno* identified from iconv (on the ASCII to EBCDIC converter) was EINVAL. EINVAL should only be encountered when the last character in the input buffer is incomplete. This did not occur. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: For information on iconv and translations, refer to *OS/390 C/C++ Programming Guide*.

Source Data Set: utility.c

Procedure Name: A2Emultiybyte_translate()

EZYTU12E UTILITY: Unexpected iconv error
errno message **rsn =** *errno2*

Explanation: The *errno* identified from iconv (on the ASCII to EBCDIC converter) is not currently checked. Unexpected error detected. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For information on iconv, refer to *OS/390 C/C++ Programming Guide*.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: A2Emultiybyte_translate()

EZYTU13E UTILITY: stilloob: select
errno message **rsn =** *errno2*

Explanation: A check on the file descriptor was made to determine if out of band data existed. An invalid value was returned. The value was less than 0. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: stilloob()

EZYTU14I UTILITY: netwrite *n* chars.

Explanation: This message is issued if -D report or -D ptydata is specified. It will write the number of characters that were read in from the socketfd (client's socket) for processing. It will be followed by hex data, preceded by the tag "Ont". This signifies a hex and ASCII representation of output data coming from the Telnet server to the net.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: netflush()

EZYTU15I UTILITY: ptywrite *n* chars.

Explanation: This message is issued if -D report or -D ptydata, is specified. It will write the number of characters which were read in from the socketfd (client's socket) for processing. It will be followed by hex data, preceded by the tag "Opt". This signifies a hex and ASCII representation of output data coming from the Telnet server to the child.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush()

EZYTU16I UTILITY: Write of masterfd.
errno message **rsn =** *errno2*

Explanation: This message is issued if -D report or -t, is specified. A negative value was received of the number of bytes written to the masterfd. Processing continues and control is returned to the calling routine.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush()

EZYTU17I UTILITY: receive suboption *sub option action*
UTILITY: send suboption *sub option action*
Valid Suboptions are:

1. **TERMINAL-TYPE**
2. **TERMINAL-SPEED**
3. **TOGGLE-FLOW-CONTROL**
4. **NAWS**
5. **SLC**
6. **LINEMODE**
7. **MODE**
8. **STATUS**
9. **X-DISPLAY-LOCATION**
10. **NEW-ENVIRON**
11. **OLD-ENVIRON**

Valid Actions are:

1. **(terminated by**
2. **, not IAC SE!)**
3. **(Empty suboption??-?)**
4. **IS -%.*s**
5. **SEND**
6. **- unknown qualifier %d (0x%x)**
7. **OFF**
8. **ON**
9. **RESTART-ANY**
10. **RESTART-XON**
11. **%d (unknown)**
12. **?%d?**
13. **%d %d (%d)**
14. **WILL**
15. **WON'T**
16. **DO**
17. **DON'T**
18. **(no option??-?)**
19. **Forward Mask**
20. **NOSUPPORT**
21. **CANTCHANGE**
22. **VARIABLE**
23. **DEFAULT**
24. **(0x%x)**
25. **(no mode??-?)**
26. **IS**
27. **SB**
28. **SE**
29. **INFO**
30. **%s (unknown)**

Explanation: A telnet negotiated suboption was either received or sent by the client. The remaining portion of this command identifies what the suboption was.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: printsub()

EZYTU18I UTILITY: write/send from NETFLUSH

Explanation: A telnet negotiated suboption was either received or sent by the client. The remaining portion of this command identifies what the suboption was.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: printsub()

EZYTU19E UTILITY: Unexpected iconv open error
errno message **rsn =** *errno2*

Explanation: The iconv_open failed for ASCII to EBCDIC converter. The *errno* identified from iconv (on the ASCII to EBCDIC converter) is not currently checked. An unexpected error was detected.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: A2Emultiybyte_translate()

EZYTU20I Int: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was input from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop()

EZYTU21I Ont: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was output from the telnet server to the client via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: netflush()

EZYTU22I Opt: *20 bytes hex data 10 char ascii***.**

Explanation: This is hex/ASCII representation of data which was output from the client to the child via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush()

EZYTU23E UTILITY: Unexpected iconv error
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. The *errno* identified from iconv (on the EBCDIC to ASCII converter) is not currently checked for. An unexpected error is detected.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultiybyte_translate()

EZYTU24E UTILITY: unexpected iconv error.
errno message **rsn =** *errno2*

Explanation: An unexpected error condition was received from the iconv call.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultiybyte_translate()

EZYTU25E UTILITY: Unexpected iconv error
errno message **rsn =** *errno2*

Explanation: An error was produced during the build of an ASCII to EBCDIC translate table from the default tables.

System Action: Processing continues.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU26E UTILITY: Unexpected iconv error
errno message **rsn =** *errno2*

Explanation: An error was produced during the build of an EBCDIC to ASCII translate table from the default tables.

System Action: Processing continues.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU27E UTILITY: Unexpected iconv open error
errno message **rsn =** *errno2*

Explanation: The iconv_open failed for EBCDIC to ASCII converter. The *errno* identified from iconv (on the EBCDIC to ASCII converter) is not currently checked for. Unexpected error detected.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: doMultiByte()

EZYTU28E UTILITY: Unexpected iconv open error
errno message **rsn =** *errno2*

Explanation: The iconv_open failed for ASCII to EBCDIC converter. The *errno* identified from iconv (on the ASCII to EBCDIC converter) is not currently checked for. Unexpected error detected.

System Action: Processing ends.

User or Operator Response: Take appropriate action based on the specified *errno* and *errno2*. For more information on iconv, refer to *OS/390 C/C++ Programming Guide*

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU29E UTILITY: ascii translation error *error*

Explanation: The iconv failed for ASCII to EBCDIC converter.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: A2Emultibyte_translate()

EZYTU30I UTILITY: A2E Begin_write len=*num_bytes***, data=***data*

Explanation: This message displays tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using the -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: A2Emultibyte_translate()

EZYTU31I UTILITY:E2A Begin_translate len=*length***,data=***data*

Explanation: The iconv failed for EBCDIC to ASCII converter. This message displays tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using the -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultibyte_translate()

EZYTU32E UTILITY:EBCDIC translation error *error*

Explanation: This message displays tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using the -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultibyte_translate()

OE TELNET Server Messages - System

EZYTY01—EZYTY14

The following are Telnet Server system messages for OS/390 TCP/IP OpenEdition.

EZYTY01E GETPTY: Out of ptys.
errno message **rsn =** *error2*

Explanation: This message indicates that all of the /dev/ptyp*XXX*s are currently in use with other sessions or processes. This message is written to the client as well as the syslog file.

System Action: Processing ends. TCP/IP continues.

User or Operator Response: Wait until some /dev/ptyp*XXX*s are freed up for use.

System Programmer Response: Check to determine if there are any ghosts or zombies that can be released.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTY02I GETPTY: open of /dev/ptyp

Explanation: These messages are only issued if -D report or -D netdata is specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTY03I gotpty: ioctl TIOCSWINSIZ

Explanation: These messages are only issued if -D report or -D netdata is specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY04E GETPTY: open error on *line*
errno message **rsn =** *error2*

Explanation: A problem occurred trying to open the corresponding /dev/ttyp*XXX* file. This message is written to the client as well as the syslog file.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* for more information regarding this failure. Possible reasons for the failure include:

1. Access is denied.
2. The process is busy.
3. The open was interrupted by a signal.
4. The system has reached the maximum number of file descriptors it can have open.
5. Permission to open is denied for one of the following reasons.
   a. The user who opened the master tty is not the same user associated with the slave tty.
   b. Internal security error.
   c. Different path name was specified for the slave than earlier opens.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY05I GETPTY: slave fd = *slavefd***, masterfd =** *masterfd*

Explanation: These messages are only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY06E gotpty: __tcsetattr
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. They correlate to the conditions of the writing for the pty.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* for more information regarding this failure.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY07T gotpty: __tcsetcp
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. A _tcsetcp was issued to inform the tty of the code pages the system uses to translate ASCII to EBCDIC and EBCDIC to ASCII. A failure occurred during the call.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* for more information regarding this failure.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY08I argv_fsum(0) = *FSUMOCLP_ARGN_NAME*

argv_fsum(1) = *FSUMOCLP_ARGN_MAGIC*
argv_fsum(2) = *FSUMOCLP_ARGN_CATNAME*
argv_fsum(3) = *FSUMOCLP_ARGN_CATSET*
argv_fsum(4) = *FSUMOCLP_ARGN_MASTERFD*
argv_fsum(5) = *FSUMOCLP_ARGN_SLAVEFD*
argv_fsum(6) = *FSUMOCLP_ARGN_HIGHFD*
argv_fsum(7) = *FSUMOCLP_ARGN_EXTRAFD*
argv_fsum(8) = *FSUMOCLP_ARGN_DEBUG*
argv_fsum(9) = *FSUMOCLP_ARGN_DEBUGW*
argv_fsum(10) = *FSUMOCLP_ARGN_10*

argv_fsum(11) = *FSUMOCLP_ARGN_TERM*
argv_fsum(12) = *FSUMOCLP_ARGN_ROWS*
argv_fsum(13) = *FSUMOCLP_ARGN_COLUMNS*
argv_fsum(14) = *FSUMOCLP_ARGN_UTMPPATH*
argv_fsum(15) = *FSUMOCLP_ARGN_UTMPNAME*
argv_fsum(16) = *FSUMOCLP_ARGN_SIGCHLD*
inherit flag = *inherit*

Explanation: This message displays the parameters that are passed to the spawned process that create the child. A few of these variables are set by the server:

argv_fsum(0) = argument name

argv_fsum(4) = master file descriptor

argv_fsum(5) = slave file descriptor

argv_fsum(8) = Debug tracing variable

argv_fsum(9) = Debug tracing variable

argv_fsum(11) = Terminal type

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTY09I login_tty: spawnp fsumoclp *child_pid*

Explanation: This message displays the pid number associated with the fork() for FSUMOCLO for cleanup.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: login_tty()

EZYTY10T login_tty: spawnp error *child_pid*
errno message **rsn =** *errno2*

Explanation: This message is written to the client as well as the syslog file. The pid number associated with the fork() for FSUMOCLO for cleanup is displayed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* and take appropriate action.

Source Data Set: telnetd.c

Procedure Name: login_tty()

EZYTY11I GETPTY: stat of /dev/ptyp

Explanation: These messages are only issued if -D report or -D netdata is specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY12I ioctl masterfd TIOCEXT

Explanation: These messages are only issued if -D report or -D netdata is specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: tty_setlinemode()

EZYTY13E login_tty failed.

Explanation: This message is written to the client as well as the syslog file. The login_tty() routine was called; this routine sets up and spawns the tty. Errors were found during processing and the caller received a return of -1.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check for further messages in the log to explain the reason for the failure.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY14E tcgetattr() failed on master tty

Explanation: A tcgetattr() was issued on the master tty, which is in fact an ioctl(). It returned with an unexpected error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the *errno* and *errno2* values for further clues on this function's failure.

Source Data Set: sys_term.h

Procedure Name: init_termbuf()

OE OSF/Motif API Messages

EZYXM01—EZYXR42

The following are OSF/Motif API messages for OS/390 TCP/IP OpenEdition. In this section, the message explanations list the function in which the error occurred. Refer to the OSF/Motif publications section in the bibliography for a list of resources that provide additional help in diagnosing and resolving the warning.

EZYXM01W The arrow direction is not correct.

Explanation: XmNarrowDirection resource in XmArrowButton widget class. XmNarrowDirection resource in XmArrowButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ArrowB.c, ArrowBG.c

EZYXM02W Incorrect resize policy.

Explanation: XmNresizePolicy resource in XmBulletinBoard widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM03W Incorrect dialog style.

Explanation: XmNdialogStyle resource in XmBulletinBoard widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM04W Incorrect shadow type.

Explanation: XmNshadowType resource in XmBulletinBoard widget class. XmNshadowType resource in XmDrawnButton widget class. XmNshadowType resource in XmFrame widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM05W Null font list (no vendor shell default).

Explanation: XmNbuttonFontList resource, XmNlabelFontList resource, or XmNtextFontList resource in XmBulletinBoard widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM11W XmCascadeButton must have correct type of XmRowColumnWidgetClass parent.

Explanation: XmCascadeButton widget class. XmCascasdeButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeBG.c

EZYXM12W Only XmMENU_PULLDOWN XmRowColumnWidgets can be submenus.

Explanation: XmCascadeButton widget class. XmCascadeButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, CascadeBG.c

EZYXM13W MapDelay must be >= 0.

Explanation: XmNmappingDelay resource in XmCascadeButton widget class. XmNmappingDelay resource in XmCascadeButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, CascadeBG.c

EZYXM14W XmCascadeButtonGadget must have xmRowColumnWidgetClass parent with rowColumnType XmMENU_PULLDOWN, XmMENU_POPUP, XmMENU_BAR or XmMENU_OPTION.

Explanation: XmCascadeButton widget class. XmCascadeButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c

EZYXM15W XtGrabPointer failed.

Explanation: XGrabPointer function. XtGrabPointer function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, MenuUtil.c, TrackLoc.c

EZYXM16W XtGrabKeyboard failed.

Explanation: XGrabKeyboard function. XtGrabKeyboard function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuUtil.c, TextF.c, TextIn.c

EZYXM21W The dialog type must be XmDIALOG_COMMAND.

Explanation: XmCommand widget class. XmNdialogType resource in XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM22W Invalid child type. The Command widget does not have this child.

Explanation: XmCommandGetChild function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM23W Invalid XmString, check for invalid charset.

Explanation: XmCommandAppendValue function. XmCommandSetValue function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM24W NULL or empty string passed in to CommandAppendValue.

Explanation: XmCommandAppendValue function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM25W mustMatch is always False for a Command widget.

Explanation: XmCommand widget class. XmNmustMatch resource in XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM26W historyMaxItems must be a positive integer greater than zero.

Explanation: XmCommand widget class. XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM31W Must call XmClipboardStartCopy() before XmClipboardCopy().

Explanation: XmClipboardCopy function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM32W Must call XmClipboardStartCopy() before XmClipboardEndCopy().

Explanation: XmClipboardEndCopy function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM33W Too many formats in XmClipboardCopy().

Explanation: XmClipboardCopy function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM41W DialogShell widget only supports one rectObj child.

Explanation: XmDialogShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DialogS.c

EZYXM42W gadgets aren't allowed in shell.

Explanation: XmDialogShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DialogS.c

EZYXM51W Margin width or height cannot be negative.

Explanation: XmDrawingArea widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DrawingA.c

EZYXM52W Margin width or height cannot be negative.

Explanation: XmNresizePolicy resource in XmDrawingArea widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DrawingA.c

EZYXM61W Fraction base cannot be zero.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM62W Incorrect form attachment type.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM63W Cannot set constraints for non-resizable widget.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM64W Attachment widget must not be null.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM65W Circular dependency in Form children.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM66W Edge attached to a widget but no widget specified.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM67W Bailed out of edge synchronization after 10,000 iterations. Check for contradictory constraints on the children of this form.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM68W Attachment widget must be have same parent as widget.

Explanation: XmForm widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM72W Only one child should be inserted in a frame.

Explanation: XmFrame widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM73W Invalid margin width.

Explanation: XmFrame widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM74W Invalid margin height.

Explanation: XmFrame widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM81W Invalid highlight thickness.

Explanation: XmGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM82W The unit type is incorrect.

Explanation: XmNunitType resource in XmGadget widget class. XmNunitType resource in XmManager widget class. XmNunitType resource in XmPrimitive widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM83W Invalid shadow thickness.

Explanation: XmGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM84W Invalid shadow thickness.

Explanation: XmNtopShadowPixmap resource in XmPrimitive widget class. XmNbottomShadowPixmap resource in XmPrimitive widget class. XmNhighlightShadowPixmap resource in XmPrimitive widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM91W Invalid XmNlabelType.

Explanation: XmNlabelType resource in XmLabel widget class. XmNlabelType resource in XmLabelGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM92W Invalid value in XmNalignment.

Explanation: XmNalignment resource in XmLabel widget class. XmNalignment resource in XmLabelGadget widget class. XmNstringDirection resource in XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM93W Invalid value in XmNstringDirection.

Explanation: XmNstringDirection resource in XmLabel widget class. XmNstringDirection resource in XmLabelGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM94W Invalid XmNlabelString - must be a compound string.

Explanation: XmNlabelString resource in XmLabel widget class. XmNlabelString resource in XmLabelGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM95W Invalid XmNacceleratorText - must be a compound string.

Explanation: XmNacceleratorText resource in XmLabel widget class. XmNacceleratorText resource in XmLabelGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXN00W List must have at least one visible item.

Explanation: XmNvisibleItemCount resource in XmList widget class. When changed, XmNvisibleItemCount must be at least 1.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN01W Invalid Selection Policy.

Explanation: XmNselectionPolicy resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN02W Invalid Size Policy.

Explanation: XmNlistSizePolicy resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN03W Invalid ScrollBar Display Policy.

Explanation: XmNscrollbarDisplayPolicy resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN04W Invalid String Direction.

Explanation: XmNstringDirection resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN05W Cannot change size policy after initialization.

Explanation: XmNlistSizePolicy resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN06W Must set item count to non-negative value.

Explanation: XmNitemCount resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN07W NULL font in SetValues ignored.

Explanation: XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN08W Invalid item(s) to delete.

Explanation: XmNitemCount resource in XmList widget class. XmListDeleteItem function. XmListDeleteItems function. XmListDeletePos function. XmListDeleteItemsPos function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN09W No Horizontal Scrollbar to set.

Explanation: XmListSetHorizPos function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN10W Invalid Margin setting.

Explanation: XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN11W Invalid Spacing Value.

Explanation: XmNlistSpacing resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN12W Cannot set items to NULL with non-zero item count.

Explanation: XmNitemCount resource in XmList widget class. XmNitems resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN13W Must set selected item count to non-negative value.

Explanation: XmNselectedItemCount resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN14W Cannot set selected items to NULL with non-zero item count.

Explanation: XmNselectedItemCount resource in XmList widget class. XmNselectedItems resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN15W Cannot set top position less than 1.

Explanation: XmNtopItemPosition resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN16W XmNitems and XmNitemCount mismatch!

Explanation: XmNitemCount resource in XmList widget class. XmNitems resource in XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN17W Cannot leave add mode in multiple selection.

Explanation: XmList widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN21W The Menu Bar cannot be changed to NULL.

Explanation: XmNmenuBar resource in XmMainWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN22W The Command Window cannot be changed to NULL.

Explanation: XmNcommandWindow resource in XmMainWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN23W Negative margin value ignored.

Explanation: XmMainWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN31W MenuShell widgets must have a xmRowColumnWidgetClass child.

Explanation: XmMenuShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN32W Attempting to manage an incomplete menu.

Explanation: XmMenuShell widget class. XmCreatePopupMenu function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN41W Invalid Dialog Type.

Explanation: XmNdialogType resource in XmMessageBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN42W Invalid Default Button Type.

Explanation: XmNdefaultButtonType resource in XmMessageBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN43W Invalid Alignment Type.

Explanation: XmNmessageAlignment resource in XmMessageBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN44W Invalid Child Type.

Explanation: XmMessageBoxGetChild function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN45W PushButton Id cannot be changed directly.

Explanation: XmMessageBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN46W Use XmNdefaultButtonType to set MessageBox default button.

Explanation: XmMessageBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN54W Invalid minimum value, must be greater than zero.

Explanation: XmNpaneMinimum resource in XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN55W Invalid maximum value, must be greater than zero.

Explanation: XmNpaneMaximum resource in XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN56W Invalid minimum/maximum value, minimum value must be smaller than the maximum value.

Explanation: XmNpaneMinimum resource in XmPanedWindow widget class. XmNpaneMaximum resource in XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN57W Constraints do not allow appropriate sizing.

Explanation: XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN58W Too few parameters.

Explanation: XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN59W Invalid 1st parameter.

Explanation: XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN61W fontList is not defined.

Explanation: XmPanedWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN71W Must be a vendor shell.

Explanation: XmAddProtocols function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN72W Protocol manager already exists.

Explanation: XmAddProtocols function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN73W More protocols than I can handle.

Explanation: XmAddProtocols function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN81W Not enough memory.

Explanation: There was not enough memory to perform the function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: PushB.c

EZYXO01W Attempt to set width to zero. Set to default value 16.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO02W Attempt to set width to zero. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO03W Attempt to set height to zero. Set to default value 16.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO04W Attempt to set height to zero. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO05W XmNhelpWidget not used by PopUps. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO06W XmNhelpWidget not used by Pulldowns. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO07W XmNhelpWidget not used by Option menus. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO08W XmNhelpWidget not used by Work Areas. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO09W Unknown value of XmNrowColumnType. It is set to WorkArea.

Explanation: XmNrowColumnType resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO10W Widget hierarchy not appropriate for this XmNrowColumnType. It is set to WorkArea.

Explanation: XmCreatePulldownMenu function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO11W Attempt to change XmNrowColumnType after initialization. The value is ignored.

Explanation: XmNrowColumnType resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO12W Unknown value of XmNorientation. The default value is used.

Explanation: XmNorientation resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO13W Attempt to set XmNorientation to unknown value. The value is ignored.

Explanation: XmNorientation resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO14W Unknown value of XmNpacking. The default value is used.

Explanation: XmNpacking resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO15W Attempt to set XmNpacking to unknown value. The value is ignored.

Explanation: XmNpacking resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO16W Unknown value of XmNentryAlignment. The default value is used.

Explanation: XmNentryAlignment resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO17W Attempt to set XmNentryAlignment to unknown value. The value is ignored.

Explanation: XmNentryAlignment resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO18W Attempt to set XmNisHomogenous to FALSE for a RowColumn widget of type XmMENU_BAR. The value is ignored.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO19W Attempt to change XmNentryClass for a RowColumn widget of type XmMENU_BAR. The value is ignored.

Explanation: XmNentryClass resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO20W Attempt to change XmNwhichButton via XtSetValues for a RowColumn widget of type XmMENU_PULLDOWN. The value is ignored.

Explanation: XmNwhichButton resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO21W Attempt to change XmNmenuPost via XtSetValues for a RowColumn widget of type XmMENU_PULLDOWN. The value is ignored.

Explanation: XmNmenuPost resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO22W Attempt to set XmNpostMenu to an illegal value. The value is ignored.

Explanation: XmNmenuPost resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO23W Attempt to change XmNshadowThickness for a RowColumn widget not of type XmMENU_PULLDOWN or XmMENU_POPUP. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO24W Attempt to change XmNorientation for a RowColumn widget of type XmMENU_OPTION. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO25W Attempt to add wrong type child to a menu (i.e. RowColumn) widget

Explanation: XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO26W Attempt to add wrong type child to a homogeneous RowColumn widget.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO27W Attempt to change XmNisHomogeneous for a RowColumn widget of type XmMENU_OPTION ignored.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO28W Tear off enabled on a shared menupane is allowed but not recommended.

Explanation: Tear-off Menus in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO29W Illegal mnemonic character. Could not convert X KEYSYM to a keycode.

Explanation: XmNmnemonic resource in XmRowColumn widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO31W The scale minimum value is greater than or equal to the scale maximum value.

Explanation: XmNminimum resource in XmScale widget class. XmNmaximum resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO32W The specified scale value is less than the minimum scale value.

Explanation: XmNvalue resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO33W The specified scale value is greater than the maximum scale value.

Explanation: XmNvalue resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO34W Incorrect orientation.

Explanation: XmNorientation resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO35W Incorrect processing direction.

Explanation: XmNprocessingDirection resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO36W Invalid highlight thickness.

Explanation: XmNhighlightThickness resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO37W Invalid scaleMultiple; greater than (max - min).

Explanation: XmNscaleMultiple resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO38W Invalid scaleMultiple; less than zero.

Explanation: XmNscaleMultiple resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO39W (Maximum - minimum) cannot be greater than INT_MAX divided by 2. Miminimum has been set to zero. Maximum may have been set to (INT_MAX/2).

Explanation: XmNmaximum resource in XmScale widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO41W The scrollbar minimum value is greater than or equal to the scrollbar maximum value.

Explanation: XmNminimum resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO42W The specified slider size is less than 1.

Explanation: XmNincrement resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO43W The specified scrollbar value is less than the minimum scrollbar value.

Explanation: XmNvalue resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO44W The specified scrollbar value is greater than the maximum scrollbar value minus the scrollbar slider size.

Explanation: XmNvalue resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO45W Incorrect orientation.

Explanation: XmNorientation resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO46W Incorrect processing direction.

Explanation: XmNprocessingDirection resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO47W The scrollbar increment is less than 1.

Explanation: XmNincrement resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO48W The scrollbar page increment is less than 1.

Explanation: XmNpageIncrement resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO49W The scrollbar initial delay is less than 1.

Explanation: XmNinitialDelay resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO50W The scrollbar repeat delay is less than 1.

Explanation: XmNrepeatDelay resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO51W Error in context manager; scrollbar backgrounds cannot be set correctly

Explanation: XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO52W Error in context manager; scrollbar foregrounds cannot be set correctly.

Explanation: XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO53W Specified slider size is greater than the scrollbar maximum value minus the scrollbar minimum value.

Explanation: XmNsliderSize resource in XmScrollBar widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO61W Invalid ScrollBar Display policy.

Explanation: XmNscrollBarDisplayPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO62W Invalid Scrolling Policy.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO63W Invalid Visual Policy.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO64W Invalid placement policy.

Explanation: XmNscrollBarPlacement resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO66W Cannot change scrolling policy after initialization.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO67W Cannot change visual policy after initialization.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO68W Cannot set AS_NEEDED scrollbar policy with a visual policy of VARIABLE.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO69W Cannot change scrollbar widget in AUTOMATIC mode.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO70W Cannot change clip window.

Explanation: XmNclipWindow resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO71W Cannot set visual policy of CONSTANT in APPLICATION_DEFINED mode.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO72W Wrong parameters passed to the function.

Explanation: XmScrollVisible function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO81W Incorrect dialog type.

Explanation: XmNdialogType resource in XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYXO82W Dialog type cannot be modified.

Explanation: XmNdialogType resource in XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYXO83W Only one work area child allowed.

Explanation: XmSelectionBox widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYXO84W Invalid child type.

Explanation: XmSelectionBoxGetChild function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYXO91W Invalid separator type.

Explanation: XmNseparatorType resource in XmSeparator widget class. XmNseparatorType resource in XmSeparatorGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Separator.c

EZYXO92W Invalid orientation.

Explanation: XmNorientation resource in XmSeparator widget class. XmNorientation resource in XmSeparatorGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Separator.c

EZYXP01W Invalid source, source ignored.

Explanation: XmNsource resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP02W Invalid edit mode.

Explanation: XmNeditMode resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP03W Text widget is editable, traversalOn must be true.

Explanation: XmNeditable resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP04W Can't find position in MovePreviousLine().

Explanation: XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextIn.c

EZYXP05W Invalid rows, must be > 0.

Explanation: XmNrows resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextOut.c

EZYXP06W XmFontListInitFontContext failed.

Explanation: XmNfontList resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c, TextOut.c

EZYXP07W XmFontListGetNextFont failed.

Explanation: XmNfontList resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c, TextOut.c

EZYXP08W Character char is not supported in font. It is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP09W Character string is not supported in font. It is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP10W Cannot use a multibyte locale without a fontset. The value is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP11W Invalid cursor position, must be >= 0.

Explanation: XmNcursorPosition resource in XmTextField widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP12W Invalid columns, must be > 0.

Explanation: XmNcolumns resource in XmText widget class. XmNcolumns resource in XmTextField widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c, TextOut.c

EZYXP13W traversalOn must always be true.

Explanation: XmTextField widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP14W Invalid columns, must be >= 0.

Explanation: XmNcolumns resource in XmTextField widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP21W Indicator type should be either XmONE_OF_MANY or XmN_OF_MANY.

Explanation: XmNindicatorType resource in XmToggleButton widget class. XmNindicatorType resource in XmToggleButtonGadget widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ToggleB.c

EZYXP31W Invalid value for navigation_type.

Explanation: XmNnavigationType resource in XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP32W Wrong value in old for navigation_type!!

Explanation: XmNnavigationType resource in XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP33W Traversal bootstrap situation with bad parameters.

Explanation: XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP34W Attempt to traverse to new tab using bad parameters.

Explanation: XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP35W startWidget is not in child list.

Explanation: XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP36W Bad parameters to TraverseToChild.

Explanation: XmManager widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP41W Invalid value for delete response.

Explanation: XmNdeleteResponse resource in VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP42W Invalid XmNpreeditType, default to OverTheSpot.

Explanation: XmNpreeditType resource in VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP45W FetchUnitType: bad widget class.

Explanation: VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP46W String to no-op conversion needs no extra parameters.

Explanation: VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP47W FetchUnitType called without a widget to reference.

Explanation: VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP51W Virtual bindings Initialize hasn't been called.

Explanation: XmVirtKeys widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: VirtKeys.c

EZYXP61W Invalid color requested from _XmAccessColorData.

Explanation: XmGetColors function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP62W Cannot allocate colormap entry for background, setting background to white.

Explanation: XmNbackground resource in Core widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP63W Cannot parse given background color, setting background to white.

Explanation: XmNbackground resource in Core widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP71W The specified Input Method failed to init : string.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP72W Cannot create the Input Method Object.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP73W XmIMFocus invoked with NULL widget.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP74W XmIMMove invoked without the Input Method focus.

Explanation: XmImGetXIM function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP81W No context found for extension.

Explanation: When trying to map the extension to a context, no context was found.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP82W _XmPopWidgetExtData: no extension found with XFindContext.

Explanation: This is a debug message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP83W XmFreeWidgetExtData is not a supported routine.

Explanation: This routine is no longer a supported interface.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP84W getLabelSecResData: not enough memory.

Explanation: There was not enough memory to perform the requested function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP91W Creating multiple XmDisplays for the same X display. Only the first XmDisplay created for a particular X display can be referenced by calls to XmGetXmDisplay.

Explanation: XmDisplay widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXP92W Received TOP_LEVEL_LEAVE with no active DragContext.

Explanation: Function requested TOP_LEVEL_LEAVE but there was no DragContext found.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXP93W Cannot set XmDisplay class to a non-subclass of XmDisplay.

Explanation: XmDisplayClass widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXQ01W The _MOTIF_DRAG_WINDOW has been destroyed.

Explanation: The window is no longer a valid context.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ02W The protocol version levels do not match.

Explanation: The level of the Motif server and client code drag protocol is not the same.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c, DragICC.c

EZYXQ03W Unable to open display.

Explanation: XmInternAtom function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ04W The atoms table is empty.

Explanation: WriteAtomsTable function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ05W The target table is empty.

Explanation: WriteTargetsTable function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ06W The target table has an inconsistent property.

Explanation: ReadTargetsTable function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ07W Invalid target table index.

Explanation: ReadTargetsTable function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ11W GenerateCallback does not expect XmCR_DROP_SITE_ENTR as a reason.

Explanation: GenerateClientCallback function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ12W Invalid selection in DropConvertCallback.

Explanation: DropConvertCallback function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ13W The drop selection was lost.

Explanation: DropLoseSelection function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ14W XGrabPointer failed.

Explanation: Call to lower level function, XGrabPointer, was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ15W ExternalNotifyHandler: the callback reason is not acceptable.

Explanation: XmDragStart function. The drag message did not contain a defined reason code.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ16W XmDragStart must be called as a result of a button press.

Explanation: XmDragStart function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ21W Unknown drag and drop message type.

Explanation: Message type received was not expected in this context.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragICC.c

EZYXQ31W No geometry specified for dragIcon pixmap.

Explanation: XmDragIcon widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ32W dragIcon created with no pixmap.

Explanation: XmNpixmap resource in XmDragIcon widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ33W String to Bitmap converter needs Screen.

Explanation: XmCvtStringToBitmap function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ41W Depth mismatch.

Explanation: XmNblendModel resource in XmDragContext widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ42W Unknown icon attachment.

Explanation: XmNattachment resource in XmDragIcon widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ43W Unknown drag state.

Explanation: GetDragIconColors function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ44W Unknown blendModel.

Explanation: XmNblendModel resource in XmDragContext widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ51W Unable to get dropSite window geometry.

Explanation: CreateAnimationSaveData function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragUnder.c

EZYXQ52W Invalid animationPixmapDepth.

Explanation: XmNanimationPixmapDepth resource in XmDropSite registry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragUnder.c

EZYXQ61W Cannot create drop sites which are children of a simple drop site.

Explanation: XmNdropSiteType resource in XmDropSite registry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ62W Receiving Motion Events without an active drag context.

Explanation: XmDragContext widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ63W Receiving operation changed without an active drag context.

Explanation: XmDragContext widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ64W Creating an active drop site with no drop procedure.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ65W Cannot set rectangles or rectangle numbers of composite drop sites.

Explanation: XmNdropRectangles resource in XmDropSite registry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ66W Registering a widget as a drop site out of sequence. Ancestors must be registered before any of their descendants are registered.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ67W Cannot register widget as a drop site more than once.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ68W DropSite type may only be set at creation time.

Explanation: XmNdropSiteType resource in XmDropSite registry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ69W Cannot change rectangles of non-simple dropsite.

Explanation: XmNdropRectangles resource in XmDropSite registry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ70W Cannot register a drop site which is a descendent of a simple drop site.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrI.c

EZYXQ71W Cannot create a discontiguous child list for a composite drop site.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrI.c

EZYXQ72W string is not a drop site child of string.

Explanation: XmDropSiteRegister function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrI.c

EZYXQ81W Geometry request to "almost" reply failed.

Explanation: The Intrinsics protocol guarantees a "Yes" response to a request with identical geometry to that which was returned by a previous request returning "almost".

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: GeoUtils.c

EZYXQ82W Invalid order found in XmSelectionBox.

Explanation: XmSelectionBox geometry.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: GeoUtils.c

EZYXQ91W Memory error.

Explanation: An attempt to allocate memory failed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Increase the region size in which the program executes.

Source Data Set: Xm.a library.

Procedure Name: Region.c

EZYXR01W Illegal representation type id.

Explanation: XmRepTypeValidValue function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RepType.c

EZYXR02W Illegal value v for rep type XmRxxx.

Explanation: XmRepTypeValidValue function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: RepType.c

EZYXR11W FetchUnitType: bad widget class.

Explanation: FetchUnitType function. Widget is not gadget, manager, or primitive.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR12W Cannot continue because of errors in a default font list.

Explanation: GetNextFontListEntry function failed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR21W Icon screen mismatch.

Explanation: XmScreen widget class.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Screen.c

EZYXR22W Cannot get XmScreen because XmDisplay was not found.

Explanation: XmGetXmScreen function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: Screen.c

EZYXR41W Could not allocate memory for color object data.

Explanation: An attempt to allocate memory failed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Increase the region size in which the program executes.

Source Data Set: Xm.a library.

Procedure Name: ColorObj.c

EZYXR42W Bad screen number from color server selection.

Explanation: GetSelection function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xm.a library.

Procedure Name: ColorObj.c

OE OSF/Motif User Interface Language (UIL) Compiler Messages

EZYXU01—EZYXU77

The following are OSF/Motif User Interface Language (UIL) compiler messages for OS/390 TCP/IP OpenEdition.

EZYXU01W The duplicate option *"string"* was ignored

Explanation: The same command line option has been repeated more than once.

System Action: The compiler continues.

User or Operator Response: Remove the duplicate command line option.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU02W The unknown option *"string"* was ignored

Explanation: An unknown option has been used in the compiler command line.

System Action: The compiler continues.

User or Operator Response: Check for a misspelled option on the command line.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU03E The additional UIL source file: *filename* was ignored

Explanation: More than one source file was specified. Only the first source file will be compiled.

System Action: The compiler continues.

User or Operator Response: Compile additional source files using separate invocations or the compiler.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU04S An error occurred opening source file: *filename*.

Explanation: The source file specified could not be opened.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c, UilSrcSrc.c

Procedure Name: various

EZYXU05S An error occurred reading next line of source file: *filename*.

Explanation: The source file specified could not be successfully read.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c, UilSrcSrc.c

Procedure Name: various

EZYXU06S An internal error occurred in: *string*

Explanation: The compiler detected an internal error.

System Action: The compiler ends.

User or Operator Response: Submit a software problem report.

System Programmer Response: None.

Source Data Set: UilDiag.c

Procedure Name: diag_handler

EZYXU07E The line was truncated at *number* characters

Explanation: The compiler encountered a source line greater than 132 characters. Characters beyond the 132 character limit were ignored.

System Action: The compiler continues.

User or Operator Response: Break each source line longer than 132 characters into several source lines. Long string literals can be created using the concatenation operator.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: src_get_source_line

EZYXU08E The value of *string* is out of range *string*.

Explanation: The value specified is outside the legal range for its type.

System Action: The compiler continues.

User or Operator Response: Change the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c, UilSarMod.c, UilSemVal.c

Procedure Name: various

EZYXU09E *string1* not terminated *string2*

Explanation: A string was not properly ended. *String1* and *string2* explain the defect.

System Action: The compiler continues.

User or Operator Response: Change the UIL module source to end the sequence properly.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: various

EZYXU10E The unprintable character *char* ignored

Explanation: The compiler encountered an illegal control character in the UIL specification file. The decimal value of the character is displayed.

System Action: The compiler continues.

User or Operator Response: Replace the character with the sequence specified in the message.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU11E The unknown sequence "*string*" ignored

Explanation: The compiler detected a sequence of printable characters it did not understand. The compiler omitted the sequence of characters shown.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU12E The unknown escape sequence \"\\char\" - \\ ignored

Explanation: A backslash was followed by an unknown escape character. The backslash is the escape character in UIL. A selected set of single characters can follow a backslash such as \n for new line. The character following the backslash was not one of the selected set.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU13E The name exceeds 31 characters.
It was truncated to: *string*

Explanation: The UIL compiler encountered a name longer than 31 characters. The compiler truncated the name to the leftmost 31 characters.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU14S The compiler ran out of virtual memory

Explanation: The compiler ran out of virtual memory.

System Action: The compiler continues.

User or Operator Response: Reduce the size of your application or increase virtual memory.

System Programmer Response: None.

Source Data Set: UilMain.c

Procedure Name: uil_mmove

EZYXU15E An unexpected string token was seen.
Parsing will resume after string.

Explanation: At the point marked in the module, the compiler found a construct, such as a punctuation mark, name, or keyword, when it was expecting a different construct. The compiler continued analyzing the module at the next occurrence of the construct stated in the message.

System Action: The compiler continues.

User or Operator Response: Check the syntax of your UIL module at the point marked by the compiler. If the module specifies case sensitive names, check that your keywords are in lowercase characters.

System Programmer Response: None.

Source Data Set: UilLexAna.c, UilLexPars.c

Procedure Name: various

EZYXU16E *string string* must be defined before this reference

Explanation: The widget pointed to in the message was either never defined or not defined prior to this point in the module. The compiler must be defined before you refer to the widget.

System Action: The compiler continues.

User or Operator Response: Check for a misspelling of the name of the widget, a missing declaration for the widget, or declaring the widget after its first reference. If names in the module are case sensitive, the spellings of the name in the declaration and in the reference must match exactly.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU17E The context requires a *string* but *string* was specified

Explanation: At the point marked in the specification, one type of object (such as a widget) is required and your specification supplied a different type of object (such as a value).

System Action: The compiler continues.

User or Operator Response: Check for misspelling. Also check to ensure that you have referred to the intended widget.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU18E *string* is not implemented yet

Explanation: You are using a feature of UIL that has not been implemented yet.

System Action: The compiler continues.

User or Operator Response: Try an alternate technique.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU19E *string* **value was found when** *string* **value was expected**

Explanation: The indicated value is not of the specific type required by UIL in this context.

System Action: The compiler continues.

User or Operator Response: Check the definition of the function or clause.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: various

EZYXU20W The *string string* **is not supported for the** *string* **object**

Explanation: Each widget or gadget supports a specific set of parameters, reasons, and children. The particular parameter, reason, or child you specified is not supported for this widget or gadget.

System Action: The compiler continues.

User or Operator Response: If a widget creation function accepts a parameter that UIL rejects, it does not necessarily mean that the UIL compiler is in error. Widget creation functions ignore parameters that they do not support without notifying you that the parameter is being ignored.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU21I This *string string* **supersedes a previous definition in this** *string string*

Explanation: A parameter or callback list has either a duplicate parameter or duplicate reason.

System Action: The compiler continues.

User or Operator Response: This is not necessarily an error. The compiler is alerting you to make sure that you intend to override the value of a prior parameter. This informational message can be suppressed using the -I option.

System Programmer Response: None.

Source Data Set: UilSemVal.c, UilSarMod.c

Procedure Name: various

EZYXU22E The name *string* **previously defined as** *string*

Explanation: The name marked by the message was used in a previous declaration. UIL requires that the names of all widgets declared within a module be unique.

System Action: The compiler continues.

User or Operator Response: Check for a misspelling. If the module is case sensitive, the spellings of names in declarations and in references must match exactly.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sem_dcl_name

EZYXU23E The value used in this context must be private

Explanation: A private value is one that is not imported or exported. In the context marked by the message, only a private value is legal. Situations where this message is issued include defining one value in terms of another, and defining parameters in terms of functions. In general, a value must be private when the compiler must know the value at compilation time. Exported values are disallowed in these contexts, even though a value is present, because that value could be overridden at run time.

System Action: The compiler continues.

User or Operator Response: Change the value to private.

System Programmer Response: None.

Source Data Set: UilSarVal.c, UilSemVal.c

Procedure Name: various

EZYXU24E The procedure *string* **was previously declared with** *number* **arguments**

Explanation: The declaration of the marked procedure specified a different number of parameters than are present in this procedure reference.

System Action: The compiler continues.

User or Operator Response: Check that you are calling the correct function. If you intend to call the procedure with a varying number of parameters, omit the argument list in the procedure declaration.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_procref_entry

EZYXU25E *string* **value was found. The argument to procedure** *string* **must be** *string* **value**

Explanation: The declaration of the marked procedure specified a different type of parameter than is present in this procedure reference.

System Action: The compiler continues.

User or Operator Response: Check that you are passing the correct parameter to the correct function. If you intend to call the procedure with varying parameter types, declare the procedure specifying 'any' as the type of the parameter.

System Programmer Response: None.

Source Data Set: various

Procedure Name: various

EZYXU26E *string string* **was found when** *string string* **was expected**

Explanation: Most parameters take values of a specific type. The value specified is not correct for this procedure.

System Action: The compiler continues.

User or Operator Response: The message indicates the expected type of parameter. Check that you have specified the intended value and that you specified the correct parameter.

System Programmer Response: None.

Source Data Set: UilSarMod.c, UilSarObj.c

Procedure Name: various

EZYXU27E *string string* **was never defined**

Explanation: Certain UIL objects, such as gadgets and widgets, can be referred to before they are defined. The marked widget is such an object, however the compiler never found the widget declaration.

System Action: The compiler continues.

User or Operator Response: Check for misspelling. If the module is case sensitive, the spellings of names in declarations and in references must match exactly.

System Programmer Response: None.

Source Data Set: UilP2Reslv.c

Procedure Name: sem_resolve_forward_ref

EZYXU28E *string string* **was already specified**

Explanation: A widget or gadget declaration can have, at most, one argument list, one callback list, and one controls list.

System Action: The compiler continues.

User or Operator Response: If you want to specify multiple lists of arguments, controls, and callbacks, you can do so within one list. For example:

```
arguments (argument_list1; argument_list2;);
```

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_save_feature

EZYXU29E *string* **item is not allowed in** *string string*

Explanation: The indicated list item is not of the type required by the list. Argument lists must contain argument entries; callback lists must contain callback entries; control lists must contain control entries; and procedure lists must contain procedure entries.

System Action: The compiler continues.

User or Operator Response: Check the syntax for the type of list entry that is required in this context and change the indicated list item.

System Programmer Response: None.

Source Data Set: UilSarObj.c, UilSemVal.c

Procedure Name: various

EZYXU30S The compilation was ended. Fix the previous errors

Explanation: Errors encountered during the compilation have caused the compiler to stop processing.

System Action: The compiler ends.

User or Operator Response: Fix the errors already diagnosed by the compiler and recompile.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_diagnostic

EZYXU31S An internal error occurred. Submit a defect report.

Explanation: The compiler diagnosed an internal error.

System Action: The compiler ends.

User or Operator Response: Get a listing and identify where the error is being issued. Try to correct any faulty syntax in this area. If you are unable to prevent this error, submit a software problem report.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_diagnostic

EZYXU33E *string* **was missing following** *string* **option**

Explanation: You used a command line option that requires a parameter and you did not provide that parameter.

System Action: The compiler continues.

User or Operator Response: Omit the option or provide the parameter.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU34S An error occurred opening the listing file: *filename*

Explanation: The compiler could not create the listing file noted in the message.

System Action: The compiler ends.

User or Operator Response: Check that you have write access to the directory you specified to hold the listing file.

System Programmer Response: None.

Source Data Set: UilLstLst.c

Procedure Name: lst_open_listing

EZYXU35S An error occurred writing to the listing file: *filename*

Explanation: The compiler could not write a line into the listing file noted in the message.

System Action: The compiler ends.

User or Operator Response: Check to see that there is adequate space in the disk specified to hold the listing file.

System Programmer Response: None.

Source Data Set: UilLstLst.c

Procedure Name: lst_output_line

EZYXU36E An invalid module structure was detected. Check the UIL module syntax

Explanation: The structure of the UIL module is incorrect.

System Action: The compiler continues.

User or Operator Response: If there are any syntax errors reported, correct them and recompile.

System Programmer Response: None.

Source Data Set: UilMain.c

Procedure Name: common_main

EZYXU37S Too many source files are open: *number*

Explanation: The compiler has a fixed limit for the number of source and include files that it can process. This number is reported in the message.

System Action: The compiler ends.

User or Operator Response: Use fewer include files.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: src_open_file

EZYXU39I errors: mv.number warnings: *number* informationals: *number*

Explanation: This message lists a summary of the diagnostics issued by the compiler and is displayed only when diagnostics have been issued.

System Action: The compiler continues.

User or Operator Response: Correct the problems reported. You can use the -I option to suppress informational and warning diagnostics that you have determined to be harmless.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_summary

EZYXU40S An error occurred opening the UID file: *filename*

Explanation: The compiler could not create the UID file noted in the message. A UID file holds the compiled user interface specification.

System Action: The compiler ends.

User or Operator Response: Check that you have write access to the directory you specified to hold the UID file.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: sem_output_uid_file

EZYXU41I No UID file was produced

Explanation: If the compiler reported recoverable or irrecoverable errors, no UID file is produced. This message informs you that the compiler did not produce a UID file.

System Action: The compiler continues.

User or Operator Response: Fix the problems reported by the compiler.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: sem_output_uid_file

EZYXU42E The creation procedure is not supported by the *string* **widget**

Explanation: You specified a creation procedure for a toolkit widget. You can specify a creation procedure only for a user defined widget.

System Action: The compiler continues.

User or Operator Response: Remove the procedure clause following the widget type.

System Programmer Response: None.

Source Data Set: various

Procedure Name: various

EZYXU43E The creation procedure is not allowed in a *string* **widget reference**

Explanation: You specified a creation procedure when referencing a widget. You can specify a creation procedure only when you declare the widget.

System Action: The compiler continues.

User or Operator Response: Remove the procedure clause following the object type.

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_verify_object

EZYXU44E The creation procedure is required in a *string* **widget declaration**

Explanation: When defining a user defined widget, you must specify the name of the creation function for creating an instance of this widget.

System Action: The compiler continues.

User or Operator Response: Insert a procedure clause following the widget type in the widget declaration. You also need to declare the creation procedure using a procedure declaration.

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_verify_object

EZYXU45W A NULL character in a string is not supported

Explanation: You have created a string that has an embedded null character. Strings are represented in a UID file and in many toolkit data structures as null ended strings. So, although the embedded nulls will be placed in the UID file, toolkit functions may interpret an imbedded null as the delimiter for the string.

System Action: The compiler continues.

User or Operator Response: Be careful when using embedded nulls.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yyparse

EZYXU46E Widget *string* **is part of a circular definition**

Explanation: The indicated object is referenced as a descendant of itself, either within its own definition or within the definition of one of the widgets in the widget tree that the object controls.

System Action: The compiler continues.

User or Operator Response: Change the definition of the indicated widget so that it is not a descendant of itself.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_callback_entry

EZYXU47S No source file was specified

Explanation: No source file was specified to compile.

System Action: The compiler ends.

User or Operator Response: Specify the name of a UIL specification file to compile.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU48W *string string* **supports only a single** *string string*

Explanation: You have specified a particular clause more than once in a context where that clause can only occur once. For example, the version clause in the module can only occur once.

System Action: The compiler continues.

User or Operator Response: Choose the correct clause and delete the others.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: various

EZYXU49W *string* **widget supports only a single control**

Explanation: The specified widget may only have one entry in its controls list.

System Action: The compiler continues.

User or Operator Response: Change the control list to have only one entry.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: out_emit_widget

EZYXU51E Place the names clause before other module clauses

Explanation: The case sensitive clause, if specified, must be the first clause following the name of the module. You have inserted another module clause before this clause.

System Action: The compiler continues.

User or Operator Response: Reorder the module clauses so that the case sensitivity clause is first.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: sar_process_module_sensitivity

EZYXU52E The color letter *string* **must be a single character**

Explanation: The string associated with each color in a color table must hold exactly one character. You have specified a string with either fewer or more characters.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_color_item

EZYXU53E The color letter was used for prior color in this table

Explanation: Each of the letters used to represent a color in a color table must be unique. If not, that letter in an icon would represent more than one color. The letter marked has been assigned to more than one color.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_color_item

EZYXU54E Row *number* **must have same width as row 1**

Explanation: The icons supported by UIL are rectangular. As a result, each of the strings used to represent a row of pixels in an icon must have the same length. The specified row does not have the same length as the first row.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_icon

EZYXU55E row *number***, column** *number***: letter "***letter***" is not in the color table**

Explanation: You have specified a color to be used in an icon that is not in the color table for that icon. The invalid color is identified in the message by displaying the letter used to represent it.

System Action: The compiler continues.

User or Operator Response: Either add the color to the color table for that icon, or use a character representing a color in the color table.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_evaluate_value_expr

EZYXU56E There are too many *string***s in** *string***. The limit is** *number*

Explanation: You exceeded a compiler limit such as the number of fonts in a font table or the number of strings in translation table. This message indicates the limit imposed by the compiler.

System Action: The compiler continues.

User or Operator Response: Restructure your UIL module.

System Programmer Response: None.

Source Data Set: UilCmd.c, UilSarVal.c

Procedure Name: various

EZYXU58W The *string* gadget is not supported. The *string* widget will be used instead

Explanation: The indicated widget type does not support a gadget variant; only a widget variant is supported for this widget type. The UIL compiler ignores the gadget indication and creates widgets of this widget type.

System Action: The compiler continues.

User or Operator Response: Restructure your UIL module.

System Programmer Response: None.

Source Data Set: UilSarMod.c, UilSarObj.c

Procedure Name: various

EZYXU59E The *string* type is not valid for *string*

Explanation: The indicated operand is not of a type that is supported by this operator.

System Action: The compiler continues.

User or Operator Response: Check the definition of the operator and make sure the type of the operand you specify is supported by the operator.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: validate_arg

EZYXU61W The *string* constraint is not supported for the *string* string

Explanation: You have specified a constraint that does not exist or is not appropriate for the constrained object.

System Action: The compiler continues.

User or Operator Response: Check for spelling errors. Check that the constraint is appropriate.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_constraint_entry

EZYXU62W Too many "*string*" options were detected, the limit is *number*

Explanation: You have specified more directories (using command line options) than the UIL compiler can process.

System Action: The compiler continues.

User or Operator Response: Reduce the number of directories specified by command line options.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU63W An error occurred while closing the source file: *filename*

Explanation: The UIL compiler was not able to properly close a source file.

System Action: The compiler continues.

User or Operator Response: Check the permissions of the file and its parent directory.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: Uil_src_cleanup_source

EZYXU64E The *string* value is circularly defined

Explanation: You attempted to declare a value using an expression that contains the value you are declaring.

System Action: The compiler continues.

User or Operator Response: Remove the circular reference from the value expression.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU65W The *string* built-in name was overridden

Explanation: You have redeclared the meaning of a nonreserved keyword. This is permitted, but generates a warning. You will not be able to use the UIL-supplied function for the keyword after it is overridden.

System Action: The compiler continues.

User or Operator Response: Make sure that you intended to override the built-in keyword. Also make sure that you no longer need the UIL-supplied function for that keyword.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sem_dcl_name

EZYXU66W The *string* argument does not support enumerated values

Explanation: You used the displayed argument along with a type that the argument does not support.

System Action: The compiler continues.

User or Operator Response: Correct the argument function. Use only supported types for the argument.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_argument_enumset

EZYXU67W The *string* argument does not support the *string* enumerated value

Explanation: You used the displayed argument along with an inappropriate value. Review the allowed values for the resources supported by the widget.

System Action: The compiler continues.

User or Operator Response: Correct the argument function. Use only supported values for the argument.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_argument_enumset

EZYXU68E The environment variable, $LANG, contains an unknown character set

Explanation: The $LANG environment variable contains an unknown character set. The UIL compiler does not have a default definition for the character set, and you have not supplied a definition using the CHARACTER_SET function.

System Action: The compiler continues.

User or Operator Response: Make sure the character set is spelled correctly. If it is a user- defined character set, make sure you have defined it within a CHARACTER_SET function.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: lex_initialize_analyzer

EZYXU69E The *string* object's controls hierarchy contains a reference to itself

Explanation: You have referred to a widget within its own controls list. Widgets cannot act as their own controlled descendants.

System Action: The compiler continues.

User or Operator Response: Remove the circular reference from the widget controls list.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU70S The value *string* is too large for context buffer

Explanation: The UIL compiler was unable to allocate an internal buffer while attempting to parse the specified value.

System Action: The compiler ends.

User or Operator Response: If possible, reduce the complexity of the value expression.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: out_emit_value

EZYXU71S Forward referencing is not allowed for *string*

Explanation: You have referenced an object before defining it.

System Action: The compiler ends.

User or Operator Response: Restructure the module so the object is defined before being referenced.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: sar_process_module_version

EZYXU72E *string* type cannot be converted to *string* type

Explanation: You have defined an expression containing two types of operands that cannot be converted into one another (for example, integer and Pixmap) or that require an explicit conversion function (for example, Boolean and Floating Point).

System Action: The compiler continues.

User or Operator Response: Make sure that the expression is correct and that the required conversion functions are used.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU73E *string* is invalid

Explanation: You defined an argument function using an argument that UIL does not allow.

System Action: The compiler continues.

User or Operator Response: Make sure the argument function is correct.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU74S An error occurred while reading the binary database

Explanation: The binary database file could not be successfully read.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: various

EZYXU75S The binary database was compiled with a future version

Explanation: The binary database was compiled by a UIL version later than the compiler version being used.

System Action: The compiler ends.

User or Operator Response: Use a binary database compiled with the current UIL compiler. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: db_incorporate

EZYXU76S An error occurred while opening the database file: *filename*

Explanation: The binary database file could not be opened.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: db_open_file

EZYXU77S An error occurred while writing to the UID file: *filename*

Explanation: The compiler could not write a line into the UID file noted in the message.

System Action: The compiler ends.

User or Operator Response: Check to see that there is adequate space in the disk specified to hold the UID file.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: various

OE X Window System Messages

EZYXW01—EZYXW62

The following are X Window System messages for OS/390 TCP/IP OpenEdition.

EZYXW01E Xlib: connection to *server_name* **refused by server**

Explanation: The specified server has refused the connection.

System Action: The application ends.

User or Operator Response: Issue the xhost command at the server to add the host address.

System Programmer Response: None.

Source Data Set: OpenDis.c

Procedure Name: XOpenDisplay

EZYXW02E Xlib: client uses different protocol version (number) than server (*server_name***)!**

Explanation: There is a mismatch between the version of the X Window System protocol used by the client library and that used by the specified server.

System Action: The application ends.

User or Operator Response: Use a server which is compatible with the client library.

System Programmer Response: None.

Source Data Set: OpenDis.c

Procedure Name: XOpenDisplay

EZYXW03E Error parsing argument "*option_name:" (value); explanation*

Explanation: The Resource Manager has been passed an invalid argument. The option name, value supplied and explanation are provided.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: ParseCmd.c

Procedure Name: _XReportParseError

EZYXW04E X connection to *device* **broken (explicit kill or server shutdown).**

Explanation: The server has been shut down or the client window has been destroyed. This message may be generated when a window is closed by the user and the application does not recognize what has happened.

System Action: The application ends.

User or Operator Response: If desired, restart the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XDefaultIOError

EZYXW05E XIO: fatal IO error *error_num* **(***error_description***) on X Server "server" after** *number* **requests (***number* **known processed) with** *number* **events remaining.**

Explanation: The connection to the server has been broken. The reported error, description of the error, and the server being used are displayed.

System Action: The application ends.

User or Operator Response: Try to restart the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XDefaultIOError

EZYXW06E Xlib: extension "*name" error_description* **on display "***address***".**

Explanation: An error has been detected by an X Windows extension. The name of the extension, description of the error and address of the server are displayed.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: extutil.c

Procedure Name: _default_exterror

EZYXW07W Warning: Current locale is not supported by Xlib

Explanation: The current locale is not supported by Xlib. The server codeset is set to ISO8859-1.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Conv.c

Procedure Name: init_iconv

EZYXW08W Warning: Unable to convert from *string1* **to** *string2***.**

Explanation: Conversion tables do not exist for converting from codeset *string1* to codeset *string2*. The default conversions are used.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Conv.c

Procedure Name: init_iconv

EZYXW09I Unable to open message catalog: X11R6.cat

Explanation: The message catalog for X Window System messages could not be opened.

System Action: Processing continues.

User or Operator Response: Verify that the NLSPATH environment variable is set to the correct value.

System Programmer Response: None.

Source Data Set: OpenDis.c, Xt/Initialize.c

Procedure Name: XOpenDisplay,XtToolitInitialize

EZYXW10E X Error of failed request: *error*
Major opcode of failed request: *number description*
(additional lines depending on X Error)
Serial number of failed request: *number*
Current serial number in output stream: *number*

Explanation: This message is displayed by the default Xlib error handler. An X Window System protocol error has been detected. The type of error detected is described. The failed request Major opcode is displayed as a number along with a description. This is followed by one or more of the following lines, depending on the type of error:

- ResourceID in failed request: *hexadecimal number*
- AtomID in failed request: *hexadecimal number*
- Value in failed request: *hexadecimal number*
- Minor code of failed request: *number*

These lines are followed by the serial number of the failed request and the current serial number.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: XlibInt.c

Procedure Name: _XPrintDefaultError

EZYXW11E XtlibError: *function_name error_type error_class description*

Explanation: An Xtlib function has detected an error. The name of the function reporting the error, the error type, the error class, and description are displayed.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: XtMsgCat.c

Procedure Name: XtErrorMsgCat

EZYXW12W XtlibWarning: *function_name error_type error_class description*

Explanation: An Xtlib function has detected a recoverable error. The name of the function reporting the error, the error type, the error class, and description are displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: XtMsgCat.c

Procedure Name: XtWarningMsgCat

EZYXW13W Xtlib: locale not supported by C library, locale unchanged.

Explanation: An attempt was made to set the current locale to a value not supported by the C library.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW14W Xtlib: locale not supported by Xlib, locale set to C

Explanation: An attempt was made to set the current locale to a value not supported by Xlib.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW15W Xtlib: X Locale modifiers not supported, using default

Explanation: The locale modifiers could not be set.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW16W XtVaTypedArg is not valid in XtVaSetSubvalues()

Explanation: XtVaSetSubvalues has been called with typed arguments.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: VarCreate.c

Procedure Name: XtVaSetSubvalues

EZYXW17W XtVaTypedArg is an invalid argument to XtVaGetSubvalues()

Explanation: XtGetSubvalues has been called with typed arguments.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: VarGet.c

Procedure Name: XtVaGetSubvalues

EZYXW18W String to BackingStore conversion needs no extra arguments

Explanation: The string to backing store conversion has been called with an extra argument.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xmu/StrToBS.c

Procedure Name: XmuCvtStringToBackingStore

EZYXW19W Xawlib: Too many parameters passed to highlight action table.

Explanation: Extra parameters were passed in a call to Highlight().

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Command.c

Procedure Name: Highlight

EZYXW20W List Widget: Unknown geometry return.

Explanation: A call to ChangeSize contained an unknown geometry request.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/List.c

Procedure Name: ChangeSize

EZYXW21W List Widget: Size changed when it shouldn't have when resizing.

Explanation: An attempt was made to resize the List widget inappropriately.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/List.c

Procedure Name: Resize

EZYXW22E MenuButton: Could not find menu widget named *name***. Error.**

Explanation: An incorrect name was passed for a menu widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: MenuButton.c

Procedure Name: PopupMenu

EZYXW23W MultiSrc: The XtNuseStringInPlace resources may not be changed.

Explanation: An attempt was made to change the XtNuseStringInPlace resource.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/MultiSrc.c

Procedure Name: SetValues

EZYXW24E *routine* **unable to allocate** *number* **bytes for widget** *name*

Explanation: The specified routine is unable to allocate the required number of bytes for a widget name.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xmu/WidgetNode.c

Procedure Name: XmuWnInitializeNodes

EZYXW25E *routine***: unable to allocate** *number1 number2* **byte widget node ptrs**

Explanation: The specified routine is unable to allocate the specified byte areas of storage for use as widget node pointers.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xmu/WidgetNode.c

Procedure Name: XmuWnFetchResources

EZYXW26W XawTextWidget: An attempt was made to insert an illegal selection.

Explanation: An attempt was made to insert an illegal selection into an Xaw Text widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: TextAction.c

Procedure Name: StartAction

EZYXW27W Xaw MultiSrc Object: possible memory leak in FreeAllPieces().

Explanation: A possible memory leak has been detected in FreeAllPieces().

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: MultiSrc.c

Procedure Name: FreeAllPieces

EZYXW28E Paned GripAction(); 1st parameter invalid.

Explanation: The first parameter passed to HandleGrip is invalid.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Paned.c

Procedure Name: HandleGrip

EZYXW29W Scrollbar Widget: Could not get geometry of thumb pixmap. Warning.

Explanation: The scrollbar widget could not get the geometry of the thumb pixmap.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Scrollbar.c

Procedure Name: CreateGC

EZYXW30W *name* **Widget: The Simple Widget class method 'change_sensitive' must be defined or inherited.**

Explanation: The specified widget does not have the Simple Widget class method 'change_sensitive' defined or inherited.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Simple.c

Procedure Name: ClassPartInitialize

EZYXW31E Paned GripAction has been passed incorrect parameters.

Explanation: HandleGrip has been passed incorrect parameters.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Paned.c

Procedure Name: HandleGrip

EZYXW32E Unknown event type in GetEventEntry().

Explanation: An unknown event type has been detected.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: SimpleMenu.c

Procedure Name: GetEventEntry

EZYXW33W No Dynamic class change of the SimpleMenu Label.

Explanation: An attempt was made to change the class of the SimpleMenu label.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: GetValues

EZYXW34W Xaw - SimpleMenuWidget: position menu action expects only one parameter which is the name of the menu.

Explanation: There should be only one parameter passed to PositionMenuAction.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenuAction

EZYXW35W Xaw - SimpleMenuWidget: could not find menu named: *name*

Explanation: The specified menu widget could not be found.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenuAction

EZYXW36W Xaw Simple Menu Widget: label *name* **is NULL or label already exists, no label is being created.**

Explanation: The specified label is NULL or the label already exists for a SimpleMenu widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: CreateLabel

EZYXW37W Xaw Simple Menu Widget: Could not find location of mouse pointer.

Explanation: The position of the mouse pointer could not be determined.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenu

EZYXW38E Xaw SmeBSB Object: Could not get Left Bitmap geometry information for menu entry "*name***"**

Explanation: The Left Bitmap geometry information for the specified menu entry could not be obtained.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW39E Xaw SmeBSB Object: Left Bitmap of entry "*name*" is not one bit deep.

Explanation: The bitmap for the specified entry is not one bit deep.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW40E Xaw SmeBSB Object: Could not get Right Bitmap geometry information for menu entry *"name"*.

Explanation: The Right Bitmap geometry information for the specified menu entry could not be obtained.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW41E Xaw SmeBSB Object: Right Bitmap of entry *"name"* is not one bit deep.

Explanation: The bitmap for the specified entry is not one bit deep.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW42W Xaw Text Widget *name*: Vertical scrolling not allowed with height resize. Vertical scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify vertical scrolling with height resizing for the specified Text widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW43W Xaw Text Widget *name*: Horizontal scrolling not allowed with wrapping active. Horizontal scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify horizontal scrolling when wrapping was active for the specified Text widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW44W Xaw Text Widget *name*: Horizontal scrolling not allowed with width resize. Horizontal scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify horizontal scrolling and width resize for the specified Text widget.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW45W Xaw Text Widget: empty selection array.

Explanation: The selection array passed to DoSelection is empty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: DoSelection

EZYXW46E Xaw Text Widget: multiply() takes exactly one argument.

Explanation: An incorrect number or arguments was passed to multiply().

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/TextAction.c

Procedure Name: Multiply

EZYXW47E Xaw Text Widget: multiply() argument must be a number greater than zero, or 'Reset'.

Explanation: The argument to multiply must be a number greater than zero, or 'Reset'

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/TextAction.c

Procedure Name: Multiply

EZYXW48W *action* This action must have only one or two parameters.

Explanation: The specified action must have only one or two parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/TextPop.c

Procedure Name: _XawTextSearch

EZYXW49W *string* **The first parameter must be either backward or forward.**

Explanation: The first parameter for the Search routine of the Text Widget is neither 'backward' or 'forward'.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/TextPop.c

Procedure Name: Search

EZYXW50W Toggle Widget Error - Attempting to create a new toggle group when one already exists.

Explanation: An attempt was made to create a new toggle group when one already exists.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Toggle.c

Procedure Name: CreateRadioGroup

EZYXW51E Aborting, due to errors resolving bindings in the Toggle widget.

Explanation: There is an error in the Toggle widget's action table.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/Toggle.c

Procedure Name: ClassInit

EZYXW52W we cannot open any input method

Explanation: Xaw is unable to open any input method.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW53W Xaw: input method doesn't support any style

Explanation: The current input method doesn't support any style.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW54W Xaw: input method doesn't support my input style.

Explanation: The input method being opened does not specify any supported style.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW55E XSMP error:

Offending minor code = *number (explanation)*
Offending sequence number = *number*
Error class = *error_class*
Severity = *severity*

(may be followed by:)

BadValue Offset = *offset*
BadValue Length = *length*
BadValue = *value*

Explanation: The session manager client routines have detected an error. The Offending minor code is displayed as a number along with an explanation. The Error class and severity are also displayed. If the Error class is BadValue, the offset, length, and the bad value are displayed.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: SM/sm_error.c

Procedure Name: _SmcDefaultErrorHandler

EZYXW56E ICE error:

Offending minor code = *number (explanation)*
Offending sequence number = *number*
Error class = *error_class*
Severity = *severity*

(may be followed by:)

BadValue Offset = *offset*
BadValue Length = *length*
BadValue = *value*

Major opcode : *number*
Reason : *reason*
Protocol name : *name*

Explanation: ICE has detected an error. Information about the error is displayed.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: ICE/error.c

Procedure Name: _IceDefaultErrorHandler

EZYXW57E X Error: *error*

Explanation: An X Protocol error has been detected. A description of the error is provided.

System Action: The application ends.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: Xmu/DefErrMsg.c

Procedure Name: XmuPrintDefaultErrorMessage

EZYXW58W XSMP error:

Offending minor code = *number (explanation)*
Offending sequence number = *number*
Error class = *error_class*
Severity = *severity*

(may be followed by:)

BadValue Offset = *offset*
BadValue Length = *length*
BadValue = *value*

Explanation: The session manager server routines have detected an error. The Offending minor code is displayed as a number along with an explanation. The Error class and severity are also displayed. If the Error class is BadValue, the offset, length, and the bad value are displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the application programming error.

Source Data Set: SM/sm_error.c

Procedure Name: _SmcDefaultErrorHandler

EZYXW60I Xlib: resource ID allocation space exhausted!

Explanation: The X Window System resource ID space has been exhausted.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XAllocID

EZYXW61I Xlib: sequence lost *number > number* **in reply type number!**

Explanation: A sequence number has been received which is less than the last received sequence number.

System Action: The application continues.

User or Operator Response: Data may have been lost, consider restarting the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XSetLastRequestRead

EZYXW62I Xlib: unexpected async reply *sequence number***!**

Explanation: An asynchronous reply has been received when none was expected.

System Action: The application continues.

User or Operator Response: Data may have been lost, consider restarting the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XAsyncReply

Configuration and Initialization Messages

EZZ0050—EZZ0345

This section contains configuration and initialization messages.

EZZ0050I *cmd* **command is not supported by this TCPIP release**

Explanation: The command is not supported by this TCPIP release. However, this command is valid and is supported by other releases of TCPIP. TCPIP rejects the command.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Check the OS/390 TCP/IP OpenEdition Configuration Guide for the list of commands that are supported by this release of TCPIP. Additionally, verify that the procedure name you specified on the command is using the TCPIP release that you expect.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0051I Error: *cmd* **command not accepted**

Explanation: TCPIP has received an unrecognized command from MVS for processing.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Note the command issued and the command response. Contact your system programmer.

System Programmer Response: Obtain a dump of the console address space. Report the error to the IBM Software Support Center.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0052I *cmd* **syntax error: Required parameter** *parm* **missing**

Explanation: The syntax for the command was not specified correctly. TCPIP detected the end of the command input before the required parameter was specified.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Determine the correct syntax for the command. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for the command syntax.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0053I Command *cmd* **completed successfully**

Explanation: The command finished without error.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0054I Error: Command *cmd* **did not complete successfully**

Explanation: The processing of the VARY TCPIP command did not complete successfully.

System Action: TCPIP continues.

User or Operator Response: Check the system log or the SYSERROR DD file specified in your TCPIP procedure for possible reasons.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0055I *cmd* **syntax error:** *parm1* **found where** *parm2* **expected**

Explanation: The syntax for the command was not specified correctly. TCPIP encountered *parm1* in the command input when *parm2* was expected.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Determine the correct syntax for the command. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for the command syntax.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0056I *cmd* **syntax error: Required value for** *parm* **is missing**

Explanation: The syntax for the command was not specified correctly. TCPIP detected the end of the command input before the required value for a parameter was specified.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Determine the correct syntax for the command. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for the command syntax.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0057I *cmd* **syntax error: Value** *val* **for** *parm* **is not valid**

Explanation: The syntax for the command was not specified correctly. The value specified for the parameter is not correct.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Determine the correct syntax for the command. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for the command syntax.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0058I *cmd* **syntax error: Value** *val* **for** *parm* **is not an integer**

Explanation: The syntax for the command was not specified correctly. The value specified for the parameter is expected to be an integer and it is not.

System Action: TCPIP continues. The command is rejected.

User or Operator Response: Determine the correct syntax for the command. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for the command syntax.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0059I *cmd* **command failed:** *reason*

Explanation: The command failed for the reason indicated

If the command you issued was VARY OBEYFILE, the following are explanations for the possible *reasons*:

- NOT AUTHORIZED - your user ID does not have proper authorization to the MVS.VARY.TCPIP.OBEYFILE RACF resource.
- SEE PREVIOUS MESSAGES - there were error messages generated while processing your profile.

If the command you issued was VARY DROP, the following are reasons for the error:

- NOT AUTHORIZED - your user ID does not have proper authorization to the MVS.VARY.TCPIP.DROP RACF resource.
- INCORRECT CONNECTION NUMBER *number* - the connection number specified cannot be dropped.
- INTERNAL COMMAND *number* FAILED ERRNO=*errno* ERRNO2=*errno2* - an internal command was issued to TCPIP.

System Action: TCPIP continues.

User or Operator Response:

For the VARY OBEYFILE command, the following are user responses for the possible *reasons*:

- NOT AUTHORIZED - make sure that your user ID has the MVS.VARY.TCPIP.OBEYFILE defined in the RACF profile and resubmit the command.
- SEE PREVIOUS MESSAGES - error messages generated while processing your profile were written to the system log. Correct the errors and resubmit the command.

For the VARY DROP command, the following are user responses for the possible *reasons*:

- NOT AUTHORIZED - make sure that your user ID has the MVS.VARY.TCPIP.DROP defined in the RACF profile and resubmit the command.
- INCORRECT CONNECTION NUMBER *number* - issue the onetstat command with the -c option to find the correct connection number. Specify a correct connection number and resubmit the command. For information about the onetstat/NETSTAT command, see *OS/390 TCP/IP OpenEdition User's Guide*.
- INTERNAL COMMAND *number* FAILED ERRNO=*errno* ERRNO2=*errno2* - An internal command was issued to TCP/IP. Report the error to the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmd

EZZ0060I PROCESSING COMMAND *cmd*

Explanation: TCPIP is about to process the command.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmds

EZZ0150I CONFIGURATION: SIGACTION() FAILED FOR *signal* **:** *reason*

Explanation: The Configuration component encountered an error attempting to setup the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

SIGABND Handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up. The Configuration component may not be automatically restarted.

SIGTERM Handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the Configuration component will not be automatically restarted when a SIGTERM is received.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0151I CONFIGURATION: RESTART SCHEDULED

Explanation: An attempt has been made to automatically restart the Configuration component following a severe error which caused the Configuration component to terminate. This message is preceded by an error message indicating why the Configuration component was terminated.

System Action: An automatic restart of the Configuration component is attempted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMMN

Procedure Name: cfTermHandler, cfAbendHandler

EZZ0152I CONFIGURATION: SHUTDOWN IN PROGRESS

Explanation: The Configuration component is permanently ending due to an error too severe to attempt an automatic restart.

System Action: The Configuration component is ended.

User or Operator Response: Contact your system programmer.

System Programmer Response: This msg is preceded by one or more error messages indicating the error that caused the Configuration component to be ended.

Source Data Set: EZACFMMN

Procedure Name: cfAbendHandler

EZZ0153I CONFIGURATION: POSSIBLE RESTART LOOP DETECTED

Explanation: An error occurred in the Configuration component that caused a restart to be attempted. The restart processing determined that the configuration component had already been restarted multiple times within a short time span and therefore ended the restart processing to prevent a restart loop from occurring.

System Action: The Configuration component is ended. In order to restart the Configuration component, TCPIP must be stopped and restarted.

User or Operator Response: This message will be preceded by several error messages which will indicate the errors that occurred in the Configuration component that caused the Configuration component to be restarted. No new configurations will be accepted until TCPIP is stopped and restarted.

System Programmer Response: This message will be preceded by several error messages which will indicate the errors that occurred in the Configuration component to cause the Configuration component to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCPIP to restart the Configuration component.

Source Data Set: EZACFMMN

Procedure Name: cfAbendHandler

EZZ0154I CONFIGURATION: UNABLE TO OPEN MESSAGE CATALOG "cfmsg.cat" : *error*

Explanation: The configuration component was unable to open the configuration message catalog "cfmsg.cat" in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

System Action: The configuration component will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the IBM C/C++ for MVS/ESA Library Reference Volume 1 for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the OpenEdition MVS Advanced Application Programming Tools publication. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0155I CONFIGURATION: UNABLE TO OPEN *ddname* **DD -** *error*

Explanation: The configuration component attempted to open the dataset specified by the indicated DD statement but was unable to open it.

System Action: The dataset will not be opened.

User or Operator Response: If use of the *ddname* DD statement is required, correct the indicated error. If writing messages to the control log using syslog() is acceptable, no action is necessary

System Programmer Response: If use of the dataset specified on *ddname* is desired, correct the indicated error. If writing messages to the control log using syslog() is acceptable, no action is necessary

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0156I INITIAL PROFILE HANDLING CONTAINED ERRORS: TERMINATING TCPIP

Explanation: An internal error occurred when handling the initial profile. This error prevents TCPIP from continuing. TCPIP has generated error messages to the system log. See these messages to determine the reason why TCPIP is being terminated.

System Action: TCPIP terminates

User or Operator Response: Error messages generated while handling the initial profile were written to the system log. Contact your system programmer.

System Programmer Response: Error messages generated while handling the initial profile were written to the system log. Report the error to the IBM Software Support Center.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0157I CONFIGURATION: *abendtext*

Explanation: The configuration component has abended. This message indicates the state of the configuration component when the abend occurred. Depending on the severity of the error, TCPIP may continue without the configuration component active or TCPIP may terminate.

System Action: TCPIP continues or is terminated.

User or Operator Response: The configuration component has abended. Contact your system programmer.

System Programmer Response: Messages were written to the system log indicating the state of the configuration component and whether TCPIP has been terminated or is still running. Use these messages to determine the state of TCPIP and to determine whether TCPIP is usable for your environment or needs to be restarted. If the errors cannot be corrected, report the error to the IBM Software Support Center.

Source Data Set: EZACFMMN

Procedure Name: doErrorRecovery, main

EZZ0158I SELECTEX FAILED: errno=*errno* **errno2=***errno2*

Explanation: An internal error occurred in TCPIP configuration processing.

System Action: The Configuration component will try to take a dump of the TCPIP address space and then abend.

User or Operator Response: Contact your system programmer.

System Programmer Response: Report the error to the IBM Software Support Center.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0300I Opened *ftype* **file** *fname*

Explanation: The file was opened for the specified type of processing.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: findInitProfile, openObeyFile, processInclude

EZZ0301I Dataset name *dsn* **is incorrect**

Explanation: The dataset name does not conform to MVS dataset naming conventions.

System Action: Configuration processing terminates for that data set.

User or Operator Response: Correct the data set name and continue.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: openObeyFile, processInclude

EZZ0302I End of File encountered

Explanation: The end of file was found before all configuration processing could be completed.

System Action: Profile processing terminates.

User or Operator Response: Correct the previous statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0303I *ftype* **file contains errors**

Explanation: The indicated file contains errors.

System Action: Configuration processing continues.

User or Operator Response: Correct the errors in the dataset and continue.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: processInclude, handleInitialProfile, processObeyFile

EZZ0304I Resuming processing of file *dsn*

Explanation: Completed processing of an include file. The original configuration file continues.

System Action: The file that contained the include statement continues.

User or Operator Response: Correct the problems in the include file and rerun.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: processInclude

EZZ0305I Cannot open file *dsn*

Explanation: The indicated file cannot be opened.

System Action: Profile processing cannot be done for the data set.

User or Operator Response: Correct the open failure and rerun the profile

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: processInclude, processObeyFile

EZZ0306I Attempted recursive include of *dsn* **on line** *linenum*

Explanation: A file cannot be included multiple times for the same configuration file processing.

System Action: The file is not included.

User or Operator Response: Correct the include statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: processInclude

EZZ0307I An internal error occurred: *reason*

Explanation: An error occurred in the configuration component that should not have occurred. For example, the configuration component could not get storage.

System Action: TCPIP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Report the error to the IBM Software Support Center.

Source Data Set: EZACFPAR

Procedure Name: handleInitialProfile

EZZ0308I Statement *statement* **on line** *linenum* **is not supported in this release**

Explanation: The statement is not supported in this release.

System Action: Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide. For more information about the statement, see the *OS/390 TCP/IP OpenEdition Configuration Guide*.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0309I Profile processing beginning for *dsn*

Explanation: Profile processing beginning.

System Action: Profile processing continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0310I File *dsn* contains no statements

Explanation: The file contains no statements.

System Action: None

User or Operator Response: Ensure that the dataset name was correctly entered and contains valid statements.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0311I The *statement* statement on line *linenum* is obsolete

Explanation: The statement is obsolete. This usually indicates that an unmodified profile from a prior release is being used.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0312I The *statement or argument* on line *linenum* contains an incorrect value *value*

Explanation: An incorrect value was specified.

System Action: Profile processing continues. However, the statement or argument is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0313I The *option* option in *statement* statement on line *linenum* is not supported in this release

Explanation: The option is not supported in this release.

System Action: Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0314I Extraneous parameter *parm* found on line *linenum*

Explanation: The parameter was used incorrectly.

System Action: The parameter is ignored.

User or Operator Response: Correct the parameter and rerun the profile.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0315I The indicated statement failed with the errno and errno2 indicated. Errnos and Errno2s are listed in OS/390 OpenEdition Messages and Codes

Explanation: One of the following may have occurred:

- Errnos for DELETE DEVICE or LINK
  - ENOLINK 1144 Device to be started did not have an associated linknam
  - EBUSY 114 For DELETE DEVICE, an associated link name still exists or the device is started.
  - EBUSY 114 For DELETE LINK, an associated home address still exists or the device is started The errno2 description in OS/390 OpenEdition Messages and Codes will give more detail into the specific reason for EBUSY.
- For START or STOP of a device
  - EOPNOTSUPP 1112 ATM device cannot be started
  - ENOENT 129 No Device name defined
  - ENOLINK 1144 No linkname defined

System Action: TCPIP continues.

User or Operator Response: Look up the errno and errno2 and correct the statement.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0316I Profile processing complete for file *dsn*

Explanation: Profile processing complete for the specified data set.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0318I *value* was found on line *linenum* and *parameter* was expected

Explanation: The expected parameter was not found.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0319I *Entry* entry value *value* is an incorrect *parameter* parameter on *statement* statement on line *linenum*

Explanation: The value is incorrect.

System Action: The value is ignored. Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0320I *Command* **value** *val* **on line** *linenum* **is too long**

Explanation: The value is too long.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0321I Internal command *number* **failed errno=***errno* **errno2=***errno2* **on line** *linenum*

Explanation: An internal error occurred in TCPIP configuration processing.

System Action: The statement fails.

User or Operator Response: Contact your system programmer.

System Programmer Response: Report the error to the IBM Software Support Center.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0322I *keyword* **value** *value* **not valid on line** *linenum***,** *replacement* **used**

Explanation: The value is incorrect, the replacement value (which may be either the default value or the closest allowed value) is used instead.

System Action: The replacement value is used. Processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0323I *Statement* **statement on line** *linenum* **had no entries**

Explanation: The statement had no entries. This may have been caused by a HOME, GATEWAY, or TRANSLATE that had a syntax error in the first entry.

If this is a HOME statement and it is the first HOME statement in a profile, the NULL HOME statement will delete all existing HOME entries. Similarly, if this is the first GATEWAY or TRANSLATE statement, all existing entries of that type are deleted. If this is not the desired effect, update the profile and rerun it.

System Action: None.

User or Operator Response: Update the file and rerun the profile.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0324I Unrecognized statement *statement* **found on line** *linenum*

Explanation: The statement is unrecognized.

This error may have been caused by a problem on the previous line or command such as an incomplete parameter information or mistyped keyword parameters.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0325I Initial profile could not be found

Explanation: The initial profile could not be found.

System Action: Profile processing terminates. TCPIP terminates

User or Operator Response: Update the files used for initial profile processing. Consult the *OS/390 TCP/IP OpenEdition Configuration Guide*.

System Programmer Response: TCPIP terminates.

Source Data Set: EZACFPAR

Procedure Name: HandleInitialProfile

EZZ0326I *statement1* **conflicts with** *statement2* **value** *value* **used on line** *linenum*

Explanation: The value is incorrect.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0327I Type *name* **on line** *linenum* **is already defined**

Explanation: The name is already defined.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0328I Type *name* **on line** *linenum* **has not been defined or has been deleted**

Explanation: The name has not been defined or has been deleted.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0329I Linkname *Linkname* **is not defined but is used for BSDRouting on line** *linenum*

Explanation: Linkname *Linkname* is not defined but is used for BSDRouting on line *linenum*

System Action: The statement is not processed. All BSDRouting statements that follow are not processed.

User or Operator Response: Define the Linkname or remove it from BSDRouting Parms.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0330I On line *linenum* **an attempt was made to define more links to device** *devname* **than is allowed by the device**

Explanation: The device specified has more links already defined than the device type supports. For example, CTC devices can have at most one link. LCS devices can have a link statement for each adapter.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0331I No home address assigned to link *linkname*

Explanation: The link was defined but no home address was given for it. No TCPIP traffic can flow over this link.

System Action: TCPIP continues.

User or Operator Response: Assign a home address for the link using the HOME statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0332I DD:PROFILE not opened. Continuing profile search

Explanation: Either there was no //PROFILE DD JCL card specified in the TCPIP cataloged procedure or the //PROFILE DD JCL card specified could not be opened.

System Action: TCPIP continues.

User or Operator Response: None

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0333I Linkname *Linkname* **is not defined with BSDROUTINGPARMS information**

Explanation: Linkname *Linkname* is not defined with BSDROUTINGPARMS information. If BSDROUTINGPARMS are used, each linkname should be defined once in the BSDROUTINGPARMS statement. The link will get default BSDROUTINGPARMS information.

System Action: TCPIP continues.

User or Operator Response: Use onetstat -d to determine if the default BSDROUTINGPARMS information is what is desired.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0334I IP forwarding is *enabled | disabled "*

Explanation: The state of IP forwarding is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0335I ICMP *will | will not "* **ignore redirects**

Explanation: The state of ICMP redirects is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0336I *A limit | No limit"* **on incoming UDP datagram queue set**

Explanation: The limit on incoming UDP datagram queue is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0337I CLAWUSEDOUBLENOP is *set | cleared "*

Explanation: The status of CLAWUSEDOUBLENOP is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0338I *TCP | UDP"* **ports 1 thru 1023** *are | are not"* **reserved**

Explanation: The status of the TCP or UDP restricted ports is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0339I More messages were found than can be reported

Explanation: The semantic processing at the end of the profile completes many configuration checks. These include LINKs without HOME statements, BSD routing checks, etc. This message is issued when more of these conditions exist than can be reported.

System Action: TCPIP continues.

User or Operator Response: Check previous error messages, correct the problems that exist and rerun the profile to receive the messages that could not be reported during this profile processing.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0340I BSDROUTINGPARMS destination address *dest_addr* **for link** *linkname* **is incorrect**

Explanation: The BSDROUTINGPARM destination address for the specified link is incorrect. If the link is a CTC or any other point to point link, the destination address must be a host address. if the link is not a point to point link, then the destination address must be a network or subnetwork address.

System Action: TCPIP continues. TCPIP ignores the statement.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0341I Configuration component cannot modify TCP/IP configuration

Explanation: A prior internal error has eliminated the ability of the configuration component to modify the TCPIP stack.

System Action: TCPIP continues.

User or Operator Response: The only way to correct the problem with the configuration component is to stop TCPIP and restart it.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0342I The DELETE PORT/PORTRANGE failed on the specified line number

Explanation: One of the following occurred:

- The port was never defined
- The port was defined, but then deleted
- The port define attributes do not agree with the delete attributes

  For example, a PORTRANGE was used to define the port, but a DELETE PORT was used to delete the port.

System Action: TCPIP continues.

User or Operator Response: Correct the statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0343I Mismatched *statement* **on line** *linenum*

Explanation: The characteristic used on the specified line conflicts with a previous definition. For example, a TRANSLATE statement specified a network address not supported by the linkname used.

System Action: TCPIP ignores the statement. TCPIP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0344I Permanent loopback address 127.0.0.1 specified on line *linenum* **cannot be added to the HOME list**

Explanation: The HOME list entry on the specified line used the permanent loopback IP address 127.0.0.1. The permanent loopback IP address 127.0.0.1 cannot be added to the HOME list.

System Action: TCPIP ignores the statement. TCPIP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0345I STOPONCLAWERROR is *enabled | disabled"*

Explanation: The status of STOPONCLAWERROR is displayed. After STOPONCLAWERROR is enabled it cannot be disabled.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0347I File I/O error occurred on input file %s

Explanation: A file I/O error occurred reading the specified input file.

System Action: Configuration processing of the input file ends.

User or Operator Response: Determine the reason for the I/O error and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPSC

Procedure Name: getData

EZZ0348I Gate list entry number *routenum* **on line** *linenum* **for destination** *destaddr* **is unreachable through interface** *intfaceaddr* **on** *interface*

Explanation: The specified gate list entry within the GATEWAY statement on the line specified is in error. The destination address is unreachable through the specified interface using any known routes.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored..

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement see the OS/390 TCP/IP OpenEdition Configuration Guide.

System Programmer Response: None.

Source Data Set: EZACFPSC

Procedure Name: getData

ONETSTAT Messages

EZZ2351—EZZ2382

This section contains onetstat messages.

EZZ2351I Incorrect option: *option***.**

Explanation: You specified an incorrect option.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the syntax of the incorrect option, and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2352I Missing parameter after option *option***.**

Explanation: You specified an *option* option without a value.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify a parameter after *option*, and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2353I Extraneous parameter *parameter***.**

Explanation: You specified an extraneous parameter.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter, and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2354I Incorrect integer parameter *parameter***.**

Explanation: You specified a parameter which is not numeric when a numeric parameter was expected.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter, and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2355I *argument* **specified for** *parameter* **must be 8 characters or less.**

Explanation: You specified an incorrect *parameter* argument.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify an *argument* less than or equal to 8 characters and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2356I More than one *option* **was specified.**

Explanation: You specified more than one *option* option.

System Action: The onetstat program halts and exits.

User or Operator Response: Remove extraneous *option* and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2357EI You specified more than the maximum of 6 select values.

Explanation: You specified more than the maximum of 6 select values.

System Action: The onetstat program halts and exits.

User or Operator Response: Remove extraneous selections and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2360I Select option *option* **can be done only with** *option***.**

Explanation: You attempted to select an option response without using one of the specified options.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify select options with a correct option and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2365I Incorrect IP address *ipaddress***.**

Explanation: You specified a incorrect IP address.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify a correct IP address and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2366I Incorrect subnet mask *subnetMask***.**

Explanation: You specified a incorrect subnet mask.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify a correct subnet mask and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2370I *ioctl* **failed with error :** *error* **(***errno***/***errno2***).**

Explanation: The onetstat issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the indicated error. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

System Programmer Response: Correct the indicated error. Refer to the *C/C++ Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDNET, EZACDNE0

EZZ2371I *devicename***, specified for link** *linkname***, was not found.**

Explanation: The onetstat command processor was parsing the TCP/IP interface table to get the -d option response information. A link record was found which had a link name of *linkname* and a device name of *devicename*, but the TCP/IP interface table did not contain a device record for that device.

System Action: The onetstat program continues.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the onetstat was processing the interface table. Retry the onetstat request. If the problem persists, check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact the IBM Software Support Center.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact the IBM Software Support Center.

Source Data Set: EZACDNET

EZZ2372I Incorrect connection number *connId***.**

Explanation: You specified an incorrect connection number.

System Action: The onetstat program halts and exits.

User or Operator Response: To find the correct connection number, issue onetstat -c. Specify a correct connection number and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNE0

EZZ2373I DROP connection process failed RACF authorization checking

Explanation: You attemped to drop a connection, but your user ID does not have MVS.VARY.TCPIP.DROP defined in the RACF profile.

System Action: The onetstat program halts and exits.

User or Operator Response: Make sure that your user ID has the MVS.VARY.TCPIP.DROP defined in the RACF profile and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNE0

EZZ2374I Unable to open message catalog "netmsg.cat" : *error*

Explanation: The onetstat was unable to open the onetstat message catalog "netmsg.cat" in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

System Action: The onetstat will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *IBM C/C++ for MVS/ESA Library Reference Volume 1* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the OpenEdition MVS Advanced Application Programming Tools publication. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACDNET

EZZ2375I sigaction() failed for *signal* **:** *reason*

Explanation: The onetstat encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the onetstat will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function	Description
SIGABND	handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.
SIGTERM	handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2376I Could not determine the TCPIPjobname, using default of 'INET'

Explanation: Onetstat called the OE service __iptcpn() to retrieve the resolver supplied TCPIPjobname and failed. Onetstat could not determine the jobname for the TCPIP stack. A default value of 'INET' will be used for TCPIPjobname.

System Action: The onetstat program continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for onetstat to communicate with a particular stack, the TCPIPjobname should be set in the appropriate resolver configuration file or data set. Make sure that the TCPIPjobname statement in the appropriate resolver configuration file or data set is correct and resubmit the onetstat command. For more information about the search order for locating the resolver configuration file or data set, see *MVS/ESA Planning: OpenEdition MVS*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifing its jobname.

Source Data Set: EZACDNET

EZZ2377I Could not establish affinity with *tcpipname* **(***error_code***/***reason***) - can not provide the requested option information**

Explanation: Onetstat called setibmopt() to associate itself with the TCPIP instance *tcpipname*, and failed with the displayed *error_code* and *reason*. The requested option information can not be provided.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the error indicated by *error_code* and *reason* and reissue the onetstat command. For onetstat to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that defined in the BPXPRM parmlib member. For more information about the customizing the BPXPRM parmlib member, see *MVS/ESA Planning: OpenEdition MVS*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifing its jobname.

System Programmer Response: Correct the error indicated by *error_code* and *reason* and reissue the onetstat command. For onetstat to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that defined in the BPXPRM parmlib member. For more information about the customizing the BPXPRM parmlib member, see *MVS/ESA Planning: OpenEdition MVS*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifing its jobname.

Source Data Set: EZACDNET

EZZ2378I Termination request is received. The program is interrupted.

Explanation: This message is displayed when a termination request is received while the onetstat command is in progress.

System Action: The onetstat program halts and exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2379I Command abend with *abendcode***,** *reasoncode***.**

Explanation: This message is displayed when an internal programming error caused onetstat program to be abnormally terminated.

System Action: The onetstat program halts and exits.

User or Operator Response: If possible, recreate the problem with -z option to collect debug information and forward the results to the TCPIP administrator.

System Programmer Response: The abend information will be written to the syslog output. Contact the IBM Software Support Center with the syslog output.

Source Data Set: EZACDNET

EZZ2380I Command option can not be used with a select-string

Explanation: You attempted to drop a connection with a select-string specified.

System Action: The onetstat program halts and exits.

User or Operator Response: Specify the command option without select-string and reissue the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2381I Incorrect option argument: *argument***.**

Explanation: You specified an incorrect option argument.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the syntax of the incorrect option argument, and resubmit the onetstat command. For information about the onetstat command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

EZZ2382I Unable to open *type* **socket to** *tcpipname***:** *error*

Explanation: The onetstat must open a UDP or RAW socket to TCP/IP in order to retrieve the requested information from TCP/IP. The onetstat's attempt to open the socket failed due to the specified error.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the problem indicated by *error*. Refer to the *C/C++ Library Reference* for further explanation of the socket errors. Be sure that TCPIP is active.

System Programmer Response: Correct the problem indicated by *error*. Refer to the *C/C++ Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDNET

OPING Messages

EZZ3100—EZZ3119

This section contains OPING messages.

EZZ3109I Extraneous parameter *parameter***.**

Explanation: You specified an extraneous parameter.

System Action: The oping program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter and resubmit the oping command. For information about the oping command, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3110I Unknown option *option***.**

Explanation: You specified an unknown option.

System Action: The oping program halts and exits.

User or Operator Response: Check the unknown option for misspellings or other problems. Correct the option and resubmit the oping command. For information about the oping command, see *OS/390 TCP/IP OpenEdition User's Guide*. Also consult related MAIN pages and/or online help, if available.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3111I Unknown host *name***.**

Explanation: The host name specified could not be resolved to an IP address.

System Action: The oping program halts and exits.

User or Operator Response:

1. Correct the syntax of the host name and resubmit the oping command. For information about the oping command, see *OS/390 TCP/IP OpenEdition User's Guide*.
2. Check that the specified host name is valid. If the host name looks correct, check with your system programmer to verify the host address.
3. Use the IP address, if it's known.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3112I Host name or address not entered.

Explanation: While attempting to submit the oping command, a requested host name or IP address was missing.

System Action: The oping program halts and exits.

User or Operator Response: Reissue the oping command with the host identification included.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3113I Missing number after *option* **option.**

Explanation: You specified an *option* option without a value.

System Action: The oping program halts and exits.

User or Operator Response: Specify an *option* value between the accepted minimum and maximum values and resubmit the oping command. For information about the oping command, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3114I The value of *option* **must be between** *minvalue* **and** *maxvalue***.**

Explanation: You specified an incorrect *option* option value.

System Action: The oping program halts and exits.

User or Operator Response: Specify an *option* value between the accepted minimum and maximum values and resubmit the PING command. For information about the oping command, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3115I Unable to open RAW socket: *error*

Explanation: The oping attempted to open a raw socket in order to issue an ICMP Echo request for the ping function. The oping was unable to open the raw socket for the reason specified by *error*.

System Action: The oping program halts and exits.

User or Operator Response: Correct the error indicated. Refer to the *C/C++ Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the error indicated. Refer to the *C/C++ Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDPIN

EZZ3116I sigaction() failed for *signal* **:** *reason*

Explanation: The oping encountered an error attempting to setup the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the oping will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function	Description
SIGABND	Handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.

SIGTERM Handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3117I Termination request is received. The program is interrupted.

Explanation: This message is displayed when a termination request is received while the oping command is in progress.

System Action: The oping program halts and exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

EZZ3118I Command abend with *abendcode*, *reasoncode*.

Explanation: This message is displayed when an internal programming error caused oping program to be abnormally terminated.

System Action: The oping program halts and exits.

User or Operator Response: If possible, recreate the problem with -d option to collect debug information and forward the results to the TCPIP administrator.

System Programmer Response: The abend information will be written to the syslog output. Contact your IBM Software Support Center with the syslog output.

Source Data Set: EZACDPIN

EZZ3119I Interaction attention request is received. The program is interrupted.

Explanation: This message is displayed when a interaction attention request is received while the oping command is in progress.

System Action: The oping program is interrupted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

SNMP Subagent Messages

EZZ3200—EZZ3220

This section contains SNMP subagent messages.

EZZ3200I SNMP subagent: Internal Error *code*

Explanation: The subagent encountered an internal programming error.

CODE	Meaning
01	mkDPIopen failed
02	failure parsing dpi packet (DPIopen)
03	no DPI response to DPI open
04	agent rejected the Open request - reason unknown
05	subagent not authorized to agent
06	DPIget_fd_for_handle failed
07	failure during mkDPIregister
08	failure parsing DPI packet (DPIregister)
09	Sever error processing packet.
0A	subagent could not get TSEB address
0B	subagent could not get TSDB address
0C	subagent could not get TSDX address
0D	subagent control block pointer was NULL

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: If possible, recreate the problem with subagent trace level 4. Contact the TCPIP administrator.

System Programmer Response: Error information will be written to the SYSLOG Daemon output file. Contact your IBM support center with the syslog output.

Source Data Set: EZASADPI.C

Procedure Name: do_connect_and_open, do_register, call_function, main

EZZ3201I SNMP subagent: duplicate subagent identifier error

Explanation: The SNMP Agent rejected the DPI open request from the subagent because another subagent has already connected to the Agent using the same subagent identifier.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: The subagent identifier for the MVS SNMP subagent is 1.3.6.1.4.1.2.6.19.3. Ensure that no user DPI programs are using this subagent identifier.

System Programmer Response: The subagent identifier for the MVS SNMP subagent is 1.3.6.1.4.1.2.6.19.3. Ensure that no user DPI programs are using this subagent identifier.

Source Data Set: EZASADPI.C

Procedure Name: do_connect_and_open

EZZ3202I SNMP SUBAGENT: INITIALIZATION COMPLETE

Explanation: The SNMP subagent has completed initialization and is ready to start processing requests.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3203I SNMP SUBAGENT: RESTART SCHEDULED

Explanation: An attempt has been made to automatically restart the SNMP subagent following a severe error which caused the SNMP subagent to be ended. This message is preceded by an error message indicating why the SNMP subagent was ended.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: exitsuba

EZZ3204I SNMP subagent: sigaction() failed for *signal* **:** *reason*

Explanation: The SNMP subagent encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function	Description
SIGABND	Handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up. The subagent may not be automatically restarted.
SIGTERM	Handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the subagent will not be automatically restarted when a SIGTERM is received.
SIGPIPE	Handler allows the subagent to detect when the connection to the SNMP agent has been terminated by the agent (for example, if the agent times out while waiting for the subagent response and closes the connection). If sigaction fails for SIGPIPE, subagent may either hang or terminate without restart.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3205I SNMP SUBAGENT: SHUTDOWN IN PROGRESS

Explanation: The SNMP subagent is permanently ending. This could be caused by a STOP command being issued for the TCPIP address space. SNMP subagent ending will also occur if the SNMP subagent encounters an error too severe to attempt automatic restart.

System Action: The SNMP subagent is ended.

User or Operator Response: If the message is issued due to a STOP command, no action is necessary. If this message occurs following an unrecoverable subagent error, it will be preceded by one or more error messages indicating the error that caused the subagent to be ended. Correct the errors listed by the previous error messages.

System Programmer Response: If the message is issued due to a STOP command, no action is necessary. If this message occurs following an unrecoverable subagent error, it will be preceded by one or more error messages indicating the error that caused the subagent to be ended. Correct the errors listed by the previous error messages. TCP/IP must be ended and restarted to restart the subagent.

Source Data Set: EZASADPI.C

Procedure Name: endsuba

EZZ3206I SNMP subagent: unable to open UDP socket to TCPIP: *error*

Explanation: The SNMP subagent must open a UDP socket to TCP/IP in order to retrieve the SNMP variable information from TCP/IP. The subagent's attempt to open the UDP socket failed due to the specified error.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: Correct the problem indicated by *error*. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

System Programmer Response: Correct the problem indicated by *error*. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3207I SNMP subagent: possible restart loop detected

Explanation: An error occurred in the SNMP subagent that caused a restart to be attempted. The restart processing determined that the subagent had already been restarted multiple times within a short time span and therefore ended the restart processing to prevent a restart loop from occurring.

System Action: The SNMP subagent is ended. In order to restart the SNMP subagent, TCP/IP must be stopped and restarted. The MIB variables provided by the subagent will be unavailable until TCP/IP is restarted.

User or Operator Response: This message will be preceded by several error messages which will indicate the errors that occurred in the SNMP subagent to cause the subagent to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCPIP to restart the SNMP subagent.

System Programmer Response: This message will be preceded by several error messages which will indicate the errors that occurred in the SNMP subagent to cause the subagent to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCP/IP to restart the SNMP subagent.

Source Data Set: EZASADPI.C

Procedure Name: exitsuba

EZZ3208I SNMP subagent: *ioctl* **failed with error:** *error*

Explanation: The subagent issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error.

System Action: The subagent is unable to process the request for the MIB variable. An error response is returned to the Agent.

User or Operator Response: Correct the indicated error. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

System Programmer Response: Correct the indicated error. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_icmp_tbl, sd_get_ifEntry_tbl, sd_get_ip_tbl, sd_get_ipAddrEntry_tbl, sd_get_ipForwardEntry_tbl, sd_get_ipNetToMediaEntry_tbl, sd_get_tcp_tbl, sd_get_tcpConnEntry_tbl, sd_get_udp_tbl, sd_get_udpEntry_tbl

EZZ3209I SNMP subagent: device *device_name***, specified for link** *link_name***, was not found**

Explanation: The SNMP subagent was parsing the TCP/IP interfaces table to build the SNMP interfaces MIB variable table. A link record was found which had a linkname of *link_name* and a device name of *device_name*, but the TCP/IP interfaces table did not contain a device record for that device.

System Action: The subagent continues processing the interfaces table, but a stack table entry will not be made for the link in error.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the subagent was processing the interfaces table. Retry the SNMP request. If the problem persists, contact the TCPIP administrator.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact your IBM Software Support Center.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ifEntry_tbl

EZZ3210I SNMP subagent: interface index *index* **not found in SNMP interfaces table**

Explanation: The SNMP subagent was parsing the TCP/IP interfaces data table to add the interface counters to the SNMP interfaces MIB records built from processing the TCP/IP interfaces table. An interface data record was found that did not have a corresponding interface record.

System Action: The subagent continues processing the interfaces data table, but the SNMP interfaces MIB table will not contain data for the listed link.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the subagent was processing the interfaces table. Retry the SNMP request. If the problem persists, contact the TCP/IP Administrator.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact your IBM Software Support Center.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ifEntry_tbl

EZZ3211I SNMP subagent: unable to open ping port: *error*

Explanation: The subagent attempted to open a raw socket in order to issue an ICMP Echo request for the remote ping function. The subagent was unable to open the raw socket for the reason specified by *error*.

System Action: The subagent will not be able to perform the remote ping request and will return an error for the remote ping MIB variable.

User or Operator Response: Correct the error indicated. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

System Programmer Response: Correct the error indicated. Refer to the "C/C++ Library Reference" for further explanation of the socket errors.

Source Data Set: EZASARPG.C

Procedure Name: mkPINGport

EZZ3212I SNMP subagent: unable to open message catalog "subamsg.cat" : *error*

Explanation: The subagent attempted to open the subagent message catalog "subamsg.cat" in the message catalog directory, but was unable to open the catalog.

System Action: The subagent will use the internal default messages instead of the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3213I SNMP subagent: waiting for group *mib_tree*

Explanation: The SNMP subagent has attempted to register the specified MIB tree, *mib_tree*, but another DPI subagent had already registered the requested MIB tree with a higher priority, or the SNMP subagent had successfully registered the specified MIB tree, but the Agent received a later registration from another subagent which requested a higher priority than that with which the SNMP subagent was currently registered.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. If the failing MIB tree becomes available at a later time the subagent will begin processing for that MIB tree.

User or Operator Response: If it is acceptable that a DPI subagent program other than the MVS SNMP subagent is providing the processing for the MIB variables in the specified MIB tree, no action is necessary. Otherwise, the other DPI subagent must be ended in order for the MVS SNMP subagent to provide the processing for the variables in the specified MIB tree.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_register, do_unreg

EZZ3214I SNMP subagent: group *mib_tree* **unregistered by Manager**

Explanation: The SNMP subagent has received a notification that the MIB tree *mib_tree* has been unregistered. This action was initiated by a request from an SNMP Manager.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. The variables under the MIB tree unregistered by the SNMP Manager will no longer be available.

User or Operator Response: If it is acceptable that the specified MIB tree is no longer available, then no action is necessary. To regain the unregistered MIB tree, the SNMP subagent must be closed by an SNMP Manager. Do this by setting the saStatus for the SNMP subagent to invalid (2). This will cause the SNMP subagent to disconnect from the Agent, then reconnect and reregister all of its supported MIB trees, including any which were previously unregistered by an SNMP Manager request.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_unreg

EZZ3215I SNMP SUBAGENT: COULD NOT ESTABLISH AFFINITY WITH *tcpip_name* **(***error_code***/***reason***)**

Explanation: The SNMP subagent attempted to use the OE socket call, setibmopt(), to associate itself with the TCP/IP instance *tcpip_name*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance under which the SNMP subagent is initializing. The setibmopt call failed with the displayed *error_code* and *reason*.

System Action: The SNMP subagent terminates.

User or Operator Response: Most likely, the TCP/IP instance's name was not defined correctly to OMVS. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure OMVS. Insure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME. Recycle OMVS or TCP/IP if a change is necessary. If none of the above error conditions exist contact your system programmer.

System Programmer Response: Correct the error indicated by *error_code* and *reason*.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3216I SNMP SUBAGENT: LOST CONNECTION TO SNMP AGENT

Explanation: The SNMP subagent was connected to the SNMP Agent, but the connection has been broken.

System Action: The subagent will try to reconnect to the SNMP Agent until successfully reconnected.

User or Operator Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the SNMP subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

System Programmer Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the SNMP subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

Source Data Set: EZASADPI.C

Procedure Name: do_open_and_register

EZZ3217I SNMP SUBAGENT: RECONNECTED TO SNMP AGENT

Explanation: The SNMP subagent has reconnected to the SNMP Agent after detecting that the prior connection had been broken.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_open_and_register

EZZ3218I SNMP SUBAGENT: CONNECTED TO OSA/SF

Explanation: The SNMP subagent has connected to the OSA/SF program.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAATM.C

Procedure Name: initAtm

EZZ3219I SNMP SUBAGENT: DISCONNECTED FROM OSA/SF

Explanation: The SNMP subagent had been connected to the OSA/SF program, but detected an error while attempting to communicate with OSA/SF. The OSA/SF connection is ended.

System Action: The subagent will try to recontact the OSA/SF program

User or Operator Response: Ensure that the OSA/SF program and ATM device are active. If possible, recreate the problem with subagent trace level 4. Contact the TCPIP administrator.

System Programmer Response: The trace messages should indicate the error which caused the subagent to disconnect from the OSA/SF program. If possible, correct the indicated error. If necessary, contact the IBM Software Support Center with the syslog output.

Source Data Set: EZASAATM.C

Procedure Name: termAtm

EZZ3220I SNMP subagent: Using loopback to connect to agent

Explanation: The SNMP subagent was unable to resolve the local host address and is using the loopback address to connect to the snmp agent instead of the host address.

System Action: The subagent will try to connect to the agent using the loopback address.

User or Operator Response: Contact the TCP/IP administrator.

System Programmer Response: Ensure that the host name is defined to the active Name server or defined in the HOSTS.SITEINFO file to allow the subagent to resolve the host name. If the loopback address is used to connect to the agent, and a password other than the snmp agent's "-c" default password is used by the subagent when connecting, then the password used by the subagent must be defined for the loopback address 127.0.0.1 in the SNMP agent's pw.src file. If the host name is defined to an active Name server, or in a HOSTS.SITEINFO file, and the subagent is still unable to resolve the name, recreate the problem with subagent trace level 4. The trace messages will indicate the error that the subagent received when attempting to resolve the host name. If possible, correct the indicated error. If necessary, contact your IBM Software Support Center with the syslog output.

Source Data Set: EZASADPI.C

Procedure Name: do_gethostname

Operator Command Related Messages

EZZ3250—EZZ3255

This section contains operator command related messages.

EZZ3250I THE MODIFY COMMAND IS NOT SUPPORTED.

Explanation: The TCPIP address space no longer supports the MODIFY command.

System Action: TCPIP continues.

User or Operator Response: A VARY operator command may be issued instead. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for assistance.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZACDMO1

EZZ3251I *job name* **IS NOT ACCEPTING VARY OR DISPLAY COMMANDS AT THIS TIME.**

Explanation: The internal state of this TCPIP instance indicates it cannot currently accept VARY or DISPLAY commands. Either TCPIP is not fully initialized, the Configuration component abnormally terminated, or TCPIP is terminating. The command request cannot be completed.

In the message text:

job name
The name of job associated with the procedure that was used to start TCPIP.

System Action: TCPIP continues.

User or Operator Response: Reissue the command. If the problem persists, save the system log and notify the system programmer.

System Programmer Response: Either TCPIP was in the process of terminating or there is a problem with the configuration component. The configuration component may either not be initialized or has abended and could not recover. If configuration cannot initialize, TCPIP will not start. Examine the system log to determine whether or not the configuration component abended and whether or not it was related to a correctable system configuration error. See the OS/390 TCP/IP OpenEdition Diagnosis Guide for assistance.

Source Data Set: TCPIP

Procedure Name: EZAO2CMD

EZZ3252I ERROR *returncode* **IN ALLOCATING STORAGE FOR CONTROL BLOCK** *cbname*

Explanation: An error occurred while attempting to allocate storage for the control block specified. The return code from the storage routine is provided.

In the message text:

returncode
The return code from the storage allocation routine.

cbname
The name of the control block for which the storage could not be obtained. The possible names are:

SATQ SNMP Subagent TrapQ block

CDMQ VARY/DISPLAY Command Request block

System Action: If the storage was for a CDMQ control block, TCPIP continues but the command process is terminated. If the storage was for a TRAPQ control block, the trap will not be generated.

User or Operator Response: Notify the system programmer of the problem.

System Programmer Response: The storage allocation attempt was for storage in common. Expanding the CSA may alleviate the problem. See the MVS Initializing and Tuning Guide for allocating more CSA.

Source Data Set: TCPIP

Procedure Name: EZACDMQ1

EZZ3253I ERROR *returncode* **IN RELEASING STORAGE FOR CONTROL BLOCK** *cbname*

Explanation: An error occurred while attempting to release storage for the control block specified. The return code from the storage routine is provided.

In the message text:

returncode
The return code from the storage routine.

cbname
The name of the control block for which the storage release failed. The possible names are:

SATQ SNMP Subagent TrapQ block

CDMQ VARY/DISPLAY Command Request block

CDMH VARY/DISPLAY Command Header block

System Action: TCPIP continues.

User or Operator Response: Notify the system programmer of the problem.

System Programmer Response: This is an internal error. Contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZACDMQ2

EZZ3255I *taskname* **HAS NOT BEEN STARTED DUE TO ERROR** *returncode* **ATTEMPTING TO** *function*

Explanation: During TCPIP initialization, a failure occurred while attempting to start the component specified.

In the message text:

taskname
The name of the task that was to be started. The possible names are:

EZACDMSM Command Manager

EZACFMMN Configuration

EZASASUB SNMP Subagent

returncode
The return code from the system call specified.

function
The function that was attempted.

System Action: TCPIP continues or ends based on the task that failed to initialize. If the task is EZACFMMN or EZACDMSM, TCPIP has not been initialized. If the task is EZASASUB, the SNMP Subagent will not be available but TCPIP is initialized.

User or Operator Response: Notify the system programmer.

System Programmer Response: Turn on ITRACE for the component specified and restart TCPIP. Gather the documentation and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZACDMIN

OSNMP Messages

EZZ3300—EZZ3327

This section contains OSNMP messages.

EZZ3300I osnmp is unable to open message catalog "snmpclim.cat" : *error*

Explanation: osnmp was unable to open the osnmp message catalog "snmpclim.cat" in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

System Action: osnmp will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the IBM C/C++ for MVS/ESA Library Reference Volume 1 for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in *OS/390 OpenEdition Programming Tools*. If the default messages are acceptable, no action is necessary.

Source Data Set: snmp

Procedure Name: main

EZZ3301I Error return from *api*

Explanation: The call to the specified routine failed. This is an internal error.

System Action: osnmp ends.

User or Operator Response: Reissue the command specifying -d 4. Collect the trace output and report the problem to the system programmer.

System Programmer Response: Check that TCPIP is running. Restart TCPIP if necessary. Make sure your system is configured correctly. See *OS/390 TCP/IP OpenEdition Configuration Guide* for configuration information. If your system is configured correctly then contact the IBM Software Support center for assistance.

Source Data Set: snmp

Procedure Name: main

EZZ3303I Error finding network address for *host*

Explanation: An IP address could not be obtained for the host specified by the -h parameter.

System Action: osnmp ends.

User or Operator Response: Verify the hostname is correct and reissue the command. If the error continues, reissue the command with the target IP address instead of the name. Notify your system programmer if the problem persists.

System Programmer Response: Determine if the name server and/or HOSTS.SITEINFO are correct See "Configuring the Domain Name System" and "Configuring the Site Table" in *OS/390 TCP/IP OpenEdition Configuration Guide*.

Source Data Set: snmp

Procedure Name: main

EZZ3304I Error finding local host name, errno = *errmsg* **Using loopback address 127.0.0.1**

Explanation: A call to gethostname() failed. The loopback IP address 127.0.0.1 will be used as the host address.

System Action: osnmp ends.

User or Operator Response: Inform the system programmer of the problem.

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.DATA data set for a valid HOSTNAME keyword. If your default TCP/IP transport provider is AnyNet, check the AnyNet ENVVAR data set for a valid HOSTNAME keyword. For more information about errnos, see *OS/390 OpenEdition Messages and Codes*.

Source Data Set: snmp

Procedure Name: main

EZZ3305I Error finding source local host address : *name*

Explanation: The IP address of the local host could not be obtained.

System Action: osnmp ends.

User or Operator Response: Inform the system programmer of the problem.

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.PROFILE data set for a valid HOMELIST specification. If your default TCP/IP transport provider is AnyNet, check the AnyNet configuration initialization procedure for a valid ISTSKIFC command.

Source Data Set: snmp

Procedure Name: main

EZZ3306I Error converting *name* **to Entity:** *errormsg*

Explanation: An error occurred when attempting to covert the destination name specified by the -h parameter to an internal destination Entity.

System Action: osnmp ends.

User or Operator Response: If the name specified by the -h parameter was not entered correctly, correct the error an reissue the command. Otherwise, there may be a configuration problem. Notify your system programmer of the problem.

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.DATA data set for a valid HOSTNAME keyword. If your default TCP/IP transport provider is AnyNet, check the AnyNet ENVVAR data set for a valid HOSTNAME keyword.

Source Data Set: snmp

Procedure Name: main

EZZ3307I Memory Allocation failed

Explanation: An allocation of C heap storage failed. There is insufficient memory to continue processing.

System Action: osnmp ends.

User or Operator Response: If the request contained a large list of variables and/or variable/value pairs, shorten the list and reissue the command.

System Programmer Response: None.

Source Data Set: snmp or wsnmpmgr

Procedure Name: various

EZZ3308I Set function requires variable name/value pair(s)

Explanation: When issuing an snmp set, a variable name and a variable value must be specified.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp -? for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3309I Only one variable allowed for *functionname* **function**

Explanation: More than one variable was specified with either the walk or the bulkwalk function. Only one is allowed.

System Action: The command ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp -? for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3310I Timeout after *number* **seconds**

Explanation: The response to the snmp request was not received before the timeout value was reached.

System Action: osnmp ends.

User or Operator Response: Reissue the command by setting a larger timeout value using the -t parameter. If the problem persists, Contact your system programmer.

System Programmer Response: Determine if the snmp agent at the target is active. Start it if necessary. Make sure that a pw.src password file has been configured. See *OS/390 TCP/IP OpenEdition Configuration Guide* for configuration information. If the password file has not been configured, configure it. The osnmpd agent must be stopped and restarted to pick up the password changes. If the problem persists, issue the osnmp command with -d 4 debug and contact the IBM Software Support Center.

Source Data Set: snmp

Procedure Name: main

EZZ3311I Option *option* **was entered without a value.**

Explanation: The option specified was entered without providing a value.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp -? for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3312I *number* **is not a valid timeout value, ignored**

Explanation: The timeout value was not valid. Either it was not a number or it was zero.

System Action: osnmp continues. The parameter is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3313I Unsupported function: *name*

Explanation: A function was entered that is not known to the command.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp -? for the command syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3314I Function *name* **requires at least one variable**

Explanation: The function specified requires at least one variable, none was entered.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp -? for the command syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3315I A *type* **request was received. The program is** *action*

Explanation: This message is displayed when either a termination signal or an interruption signal was received from LE/370 while osnmp was executing.

System Action: osnmp ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: termHand

EZZ3316I Command abended with *abendcode*, *reasoncode*.

Explanation: This message is displayed when a abend occurred while the osnmp command was in progress.

System Action: osnmp ends.

User or Operator Response: Correct the error indicated by *abencode* and *reasoncode* and reissue the command.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: abndHand

EZZ3317I sigaction() failed for *signal* **:** *reason*

Explanation: osnmp encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, osnmp will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function	Description
SIGABND	Handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.
SIGTERM	Handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.
SIGINT	Handler controls cleanup of resources during interactive attention. If sigaction fails for SIGINT, certain resources may not be properly cleaned up when a SIGINT is received.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3318I Unrecognized SMI type : *type*

Explanation: osnmp encountered value type *type* that is not in the osnmp list of SMI types. The unknown variable can come from two different sources:

```
The /etc/mibs.data file and the compiled MIB (unknown in both)
                    OR
The PDU received from the SNMP agent.
```

System Action: osnmp ends.

User or Operator Response: Based on the type received, determine if this is a valid SMI type. If it is not, first correct any incorrect entry in /etc/mibs.data. If this does not correct the problem, trace the SNMPD agent and report the problem to your system programmer. See *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: snmp

Procedure Name: SnmpPrintValue

EZZ3319I *variable* **is not a valid variable.instance**

Explanation: The syntax of a variable name or ObjectID is not correct or a variable name, not an instance of the variable was entered. The correct syntax is as follows:

Function	Example
varName.n	sysDescr.0
n.n....n	1.3.6.1.2.1.1.1.0

Where:

Variable	Description
varName	Alphanumeric, starts with an alphabetic (a-z)
n	One or more digits.

System Action: osnmp ends.

User or Operator Response: Reissue the command with the correct syntax or instance.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3320I variable *variable* **is not found in any local MIB**

Explanation: The variable specified was not found in any of the mibs available

System Action: The command ends.

User or Operator Response: If the variable specified is incorrect, correct it and reissue the command. If the variable is correct, reissue the command specifying -d 1. Collect the trace output and report the problem to the system programmer.

System Programmer Response: Determine if all the MIBs are available on the system. See *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3321I Error in /etc/mibs.data file, *lineno* **:** *error*

Explanation: While the command processor was reading the /etc/mibs.data file, it encountered an error in the file on the line specified. The possible errors are:

Value	Description
line too long	Each line in the file must be no longer than 2048. The line specified is longer.
missing field(s)	One or more fields are missing from the the file. The format of the file is: `shortname objectidentifier type`
unrecognized type	The value in the "type" field is not one of the expected types. See the *OS/390 TCP/IP OpenEdition User's Guide* for a list of types.

System Action: The line is ignored and osnmp continues.

User or Operator Response: Correct the error indicated and reissue the command.

System Programmer Response: None.

Source Data Set: snmp_mtable

Procedure Name: main

EZZ3322I *value* **is not numeric**

Explanation: The value must be numeric.

System Action: The command ends.

User or Operator Response: Correct the syntax and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3323I *value* **value is out of range**

Explanation: The value entered is outside the range defined by the SMI type.

System Action: The command ends.

User or Operator Response: Correct the syntax and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3324I *value* **is not a valid IP address**

Explanation: The IP address specified is not syntactically correct.

System Action: The command ends.

User or Operator Response: Correct the value and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3325I *value* **is not a valid OID**

Explanation: The OID specified is not syntactically correct.

System Action: The command ends.

User or Operator Response: Correct the value and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3326I Variables cannot be specified with the TRAP function

Explanation: A variable was specified after the TRAP function. Variables are not applicable when specifying TRAP.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3327I No valid PDUs returned on this *function* **request**

Explanation: A request was made to walk a MIB tree via a walk or a bulkwalk request. The next variable in the tree did not have the same prefix as the variable specified on the command. The PDU was thrown away.

System Action: osnmp ends.

User or Operator Response: Reissue the command specifying a valid MIB tree prefix.

System Programmer Response: None

Source Data Set: snmp

Procedure Name: main

Initialization Messages

EZZ4200—EZZ4216

This section contains initialization messages.

EZZ4200I TCP/IP INITIALIZATION COMPLETE FOR *jobname*

Explanation: Initialization has completed successfully for TCP/IP.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

System Action: The TCP/IP has initialized.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBTIINI

EZZ4201I TCP/IP TERMINATION COMPLETE FOR *jobname*

Explanation: This TCP/IP has terminated.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

System Action: The TCPIP procedure terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBTIINI

EZZ4202I OPENEDITION-TCP/IP CONNECTION ESTABLISHED FOR *jobname*

Explanation: This TCP/IP has established a connection with OpenEdition.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

System Action: The TCPIP procedure established a connection with OE.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBTIPFS

EZZ4203I OPENEDITION-TCP/IP CONNECTION ERROR FOR *jobname-function name,function type,return value,return code,reason code*

Explanation: This TCP/IP has failed to established a connection with OpenEdition.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

function name
The name of the function that failed while trying to establish the connection with OE.

function type
The particular type for this function.

return value
The return value from the function attempted.

return code
The return code from the function attempted.

reason code
The reason code from the function attempted.

System Action: The TCPIP has failed to initialize.

User or Operator Response: Contact the system programmer.

System Programmer Response: There are a few customization errors that could cause TCP/IP to not to be able to connect with OpenEdition. The TCP/IP jobname was not specified correctly and does not match one in the BPXPRMxx member that OpenEdition is using. OpenEdition was not configured with a CINET definition. The userid associated with the TCP/IP started procedure does not have the proper OpenEdition authorization. Collect any supporting documentation and dumps, if available, and contact the IBM Software Support.

Source Data Set: TCPIP

Procedure Name: EZBTIPFS

EZZ4204I TCPIP INITIALIZATION FOR *jobname* **FAILED.**

Explanation: Initialization failed for TCP/IP.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

System Action: This TCP/IP failed to initialize.

User or Operator Response: This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

System Programmer Response: This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

Source Data Set: TCPIP

Procedure Name: EZBTIINI

EZZ4205I TCPIP INITIALIZATION UNABLE TO START *jobname* **REASON: {TCPIP ALREADY EXISTS|MAXIMUM OF 8 TCPIPS ALREADY STARTED}.**

Explanation: This TCPIP could not be started because one was already started with the same name, or the maximum number of 8 TCPIPs were already started.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

TCPIP ALREADY EXISTS
A TCPIP with this jobname has already been started on this system.

MAXIMUM OF 8 TCPIPS ALREADY STARTED
The maximum number of TCP/IPs have already been started on this system.

System Action: This TCP/IP failed to initialize.

User or Operator Response: Determine if the correct TCPIP jobname was used to start TCPIP.

System Programmer Response: Take the action based on the specific failure explanation.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4206I TCPIP INITIALIZATION FOR *jobname* **UNABLE TO OBTAIN CSM STORAGE. STORAGE TYPE: {DATASPACE|ECSA} RETURN CODE:** *return code* **REASON CODE:** *reason code*

Explanation: TCP/IP initialization failed trying to obtain CSM storage.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

DATASPACE
TCPIP failed to obtain CSM dataspace storage.

ECSA
TCPIP failed to obtain CSM ECSA storage.

return code
The return code from the IVTCSM REQUEST(CREATE_POOL) macro invocation

reason code
The reason code from the IVTCSM REQUEST(CREATE_POOL) macro invocation

System Action: This TCP/IP will fail to initialize.

User or Operator Response: Determine from the return and reason codes from the CSM invocation as to why TCPIP was unable to obtain the storage. See Programming for CSM (SC31-8420) for return code documentation.

System Programmer Response: Determine why the storage could not be obtained. Make sure that VTAM has been started and is at the appropriate level.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4207I TCPIP INITIALIZATION FOR *jobname* **ENCOUNTERED AN ENVIRONMENT ERROR. {NOT RUNNING AS A STARTED PROCEDURE.|NOT RUNNING IN KEY 6}**

Explanation: TCP/IP initialization failed because it detected it was not running in the correct environment.

In the message text:

jobname
The jobname associated with the procedure used to start TCPIP.

NOT RUNNING AS A STARTED PROCEDURE.
TCPIP detected it was not running as a started procedure.

NOT RUNNING IN KEY 6
TCPIP detected it was not running in the correct key.

System Action: This TCP/IP will fail to initialize.

User or Operator Response: Only try to run TCP/IP as a started procedure. Make sure there is a correct entry in the Program Property Table. See the *OS/390 TCP/IP OE Program Directory* for the contents of the required PPT entry for EZBTCPIP.

System Programmer Response: Make sure TCP/IP is installed correctly and the started proc is correct. Also make sure the PPT entry is correct.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4208I BPX1SDD FAILED WITH RETURN CODE: *return code***, REASON CODE:** *reason code***. THE SNMP SUBAGENT IS NOT AVAILABLE.**

Explanation: A call to OE services routine BPX1SDD failed with the return and reason code provided. The SNMP subagent cannot be initialized and will not be available.

In the message text:

return code
The return code from BPX1SDD.

reason code
The reason code from BPX1SDD.

System Action: This TCP/IP continues.

User or Operator Response: Try and determine why BPX1SDD failed from the return code and the reason code provided. This information is found in the Assembler Callable Services for OpenEdition for MVS publication.

System Programmer Response: Contact your IBM support center for assistance.

Source Data Set: TCPIP

Procedure Name: EZBTIINI

EZZ4209I CTRACE OPTION *ctrace option* **IS NOT VALID**

Explanation: In response to the TRACE CT command one of the options value is not supported by TCPIP.

In the message text:

ctrace option
The unrecognized CTRACE option.

System Action: The CTRACE request is terminated with message ITT004I.

User or Operator Response: Reissue the TRACE CT command and specify a valid option value. Supported trace options can be found in the *OS/390 TCP/IP OpenEdition Diagnosis Guide*.

Source Data Set: TCPIP

Procedure Name: EZBCTSSM

EZZ4210I CTRACE DEFINE FAILED FOR *parmlib member name*, **RETURN CODE:** *return code* **REASON CODE:** *reason code*

Explanation: The request to define a component trace for TCPIP failed. The return codes and reason codes are defined in: *MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALESERV-DYNALLOC)*.

In the message text:

parmlib member name
The CTRACE Parmlib member name.

return code
The return code from the CTRACE DEFINE macro invocation.

reason code
The reason code from the CTRACE DEFINE macro invocation.

System Action: If the parmlib member was not CTIEZB00, then a CTRACE DEFINE is attemted using CTIEZB00. If the parmlib member was CTIEZB00, then the CTRACE DEFINE is attempted with system defaults.

User or Operator Response: Contact the system programmer to correct the parmlib member.

System Programmer Response: Correct the parmlib member and restart TCPIP or issue the `TRACE ON,COMP=SYSTCPIP,SUB=`*`procedure_name`*`,PARM=`*`parmlib_member`* command to update the component trace functions.

Source Data Set: TCPIP

Procedure Name: EZBCTINI

EZZ4211I BLDL FOR MODULE EZBITCOM FAILED, RETURN CODE: *return code*

Explanation: A BLDL macro call to locate module EZBITCOM failed with the return code provided. TCP/IP can not complete its initialization.

In the message text:

return code
The return code from the BLDL macro.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try and determine why the BLDL failed from the return code provided. This information is found in the *DFSMS/MVS Macro Instructions for '.'Data Sets* publication. Verify that load module EZBITCOM exists in data set hlq.SEZALINK, where hlq is your installation defined high level qualifier. You should also verify that hlq.SEZALINK is either in the default MVS link list or that it is explicitly specified as a STEPLIB DD card on the started procedure JCL used to start this TCP/IP instance. If no problems are found, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4212I LOAD FOR MODULE EZBITCOM FAILED, ABEND CODE: *abend code* **ABEND REASON CODE:** *abend reason code*

Explanation: A LOAD macro call for module EZBITCOM failed with the abend code and reason code provided. TCP/IP can not complete its initialization.

In the message text:

abend code
The Abend Code from the LOAD macro.

abend reason code
The abend reason code from the LOAD macro invocation.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try and determine why the LOAD failed from the abend code and abend reason code provided. This information is found in the *MVS System Codes* publication. Verify that load module EZBITCOM exists in data set hlq.SEZALINK, where hlq is your installation defined high level qualifier. You should also verify that hlq.SEZALINK is either in the default MVS link list or that it is explicitly specified as a STEPLIB DD card on the started procedure JCL used to start this TCP/IP instance. If no problems are found, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4213I COULD NOT OBTAIN CSA STORAGE, RETURN CODE: *return code*

Explanation: A STORAGE (OBTAIN) system call for CSA storage failed with the return code provided. TCP/IP can not complete its initialization.

In the message text:

return code
The return code from the STORAGE (OBTAIN) macro.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try and determine why the STORAGE (OBTAIN) call failed from the return code provided. This information is found in the *MVS Authorized Assembler Services Reference* publication. If the return code indicates that insufficient storage or system resources were available, correct the problem and restart TCP/IP. Otherwise, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4214I STORAGE RELEASE FOR CSA FAILED, RETURN CODE: *return code*

Explanation: An unexpected internal error occured while trying to free CSA storage. The return code from the STORAGE (RELEASE) system service is provided.

In the message text:

return code
The return code from the STORAGE (RELEASE) macro.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITINI

EZZ4215I TCP/IP ABEND - DUMPING

Explanation: TCP/IP recovery has been entered and a dump will be taken.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITDRC

EZZ4216I TCP/IP ABEND, DUMP SUPPRESSED DUE TO A PREVIOUS ERROR

Explanation: TCP/IP recovery has been entered, but no dump will be taken due to a previous abend by this work unit.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: TCPIP

Procedure Name: EZBITDRC

Interface Layer Messages

EZZ4300—EZZ4315

This section contains Interface Layer messages.

EZZ4300I CLAW DEVICE *device name*: **INCORRECT ADAPTER MICROCODE VERSION** *version*

Explanation: An incorrect version of the microcode was loaded onto the CLAW adapter in the workstation.

In the message text:

device name
The name of the device.

version
The version level of the adapter microcode.

System Action: TCPIP does not start the CLAW device.

User or Operator Response: Inform the system programmer about the error.

System Programmer Response: Obtain the correct level of the CLAW microcode and restart the CLAW adapter. Restart the CLAW device.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4301I CLAW DEVICE *device name*: **RECEIVED HOST NAME** *received host name* **ADAPTER NAME** *received adapter name*, **EXPECTED HOST NAME** *expected host name* **ADAPTER NAME** *expected adapter name*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters.

In the message text:

device name
The name of the device.

received host name
The host name received from the CLAW device.

received adapter name
The adapter name received from the CLAW device.

expected host name
The host name expected from the CLAW device.

expected adapter name
The adapter name expected from the CLAW device.

System Action: TCPIP does not start the CLAW device.

User or Operator Response: Inform the system programmer about the error.

System Programmer Response: Correct the host and adapter name parameters in either the CLAW adapter configuration or the CLAW DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4302I CLAW DEVICE *device name*: **ADAPTER RECEIVE SIZE OF** *adapter receive size* **IS SMALLER THAN HOST TRANSMIT SIZE OF** *host transmit size*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters. The write buffer size on the CLAW DEVICE statement is larger than the CLAW adapter receive buffer size.

In the message text:

device name
The name of the device.

adapter receive size
The size of data that the CLAW device can receive.

host transmit size
The size of data that TCP/IP can send to the CLAW device.

System Action: TCPIP does not start the CLAW device.

User or Operator Response: Inform the system programmer about the error.

System Programmer Response: Correct the buffer size in either the CLAW adapter configuration or the CLAW DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4303I CLAW DEVICE *device name*: **ADAPTER TRANSMIT SIZE OF** *adapter transmit size* **IS GREATER THAN HOST RECEIVE SIZE OF** *host receive size*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters. The read buffer size on the CLAW DEVICE statement is smaller than the CLAW adapter transmit buffer size.

In the message text:

device name
The name of the device.

adapter transmit size
The size of data that the CLAW device can send.

host receive size
The size of data that TCP/IP can receive from the CLAW device.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

System Action: TCPIP does not start the CLAW device.

User or Operator Response: Inform the system programmer about the error.

System Programmer Response: Correct the buffer size in either the CLAW adapter configuration or the CLAW DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

EZZ4305I UNABLE TO RECOVER DEVICE *device name*

Explanation: Recovery attempts for the device were unsuccessful. The original error (which initiated the recovery action) is described in message EZZ4310I. Either message EZZ4306I or EZZ4307I will also accompany this message, describing why recovery attempts were abandoned.

In the message text:

device name
The name of the device.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I.

System Programmer Response: Perform the action described for EZZ4310I.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4306I REASON: REACHED UNSUCCESSFUL RETRY THRESHOLD

Explanation: This message accompanies EZZ4305I and EZZ4310I. Recovery attempts were unsuccessful at returning the device to the Ready state.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I.

System Programmer Response: Perform the action described for EZZ4310I.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4307I REASON: ERROR ENCOUNTERED AFTER REACTIVATION

Explanation: This message accompanies EZZ4305I and EZZ4310I. Recovery attempts were successful at returning the device to the Ready state, but an error was detected on the device immediately following reactivation.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I.

System Programmer Response: Perform the action described for EZZ4310I.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4308I ERROR: CODE=*error code* **DURING** *link control function* **DEVICE** *device name***. DIAGNOSTIC CODE:** *internal diagnostic code*

Explanation: The Link Layer has detected an error during activation of the device.

In the message text:

error code
The Link-Layer Error Code.

link control function
The function being performed on the device.

device name
The name of the device.

internal diagnostic code
Internal Diagnostic for use by IBM.

System Action: TCP/IP does not start the device.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes chapter in VTAM Codes (GC31-8369), for a description of the Link Layer Error Code. If applicable, correct the hardware problem and restart the device.

System Programmer Response: Perform the action described in "VTAM Codes" for the indicated status code.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4309I ATTEMPTING TO RECOVER DEVICE *device name*

Explanation: An error was detected on the device, as reported by message EZZ4310I, and the link layer is attempting to recover the device. If the recovery attempt is successful, either message EZZ4313I or EZZ4314I will accompany this message. If the recovery attempt is unsuccessful, message EZZ4305I will be issued.

In the message text:

device name
The name of the device.

System Action: TCP/IP attempts to recover the device.

User or Operator Response: If accompanied by message EZZ4305I, device recovery was unsuccessful. Perform the action described for EZZ4310I. If accompanied by EZZ4313I or EZZ4314I, recovery of the device was successful, and no further action is necessary. If neither EZZ4305I nor EZZ4313I/EZZ4314I is issued, TCP/IP is awaiting a "ready" indication from the device. (This is typically seen on CTC devices, where the remote TCP/IP image is either down or has not yet issued a START DEV for the CTC connection.)

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4310I ERROR: CODE=*error code* **REPORTED ON DEVICE** *device name***. DIAGNOSTIC CODE:** *internal diagnostic code*

Explanation: The Link Layer has detected an error during operation of the device. The error was reported asynchronous to execution of any Link Control function.

In the message text:

error code
The Link-Layer Error Code.

device name
The name of the device.

internal diagnostic code
Internal Diagnostic for use by IBM.

System Action: The device is left in the Not Ready state.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes chapter in VTAM Codes (GC31-8369), for a description of the Link Layer Error Code. If applicable, correct the hardware problem and restart the device.

System Programmer Response: Perform the action described in VTAM Codes for the indicated status code.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4313I INITIALIZATION COMPLETE FOR DEVICE *device name*

Explanation: TCPIP has successfully started the device.

In the message text:

device name
The name of the device.

System Action: The device is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4314I INITIALIZATION COMPLETE FOR DEVICE *device name*, **LINK** *link name*

Explanation: TCPIP has successfully started the specified link for the specified device.

In the message text:

device name
The name of the device.

link name
The name of the link.

System Action: The link is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4315I DEACTIVATION COMPLETE FOR DEVICE *device name*

Explanation: TCPIP has deactivated the device.

In the message text:

device name
The name of the device.

System Action: The device and its link(s) are no longer available for use with TCPIP.

User or Operator Response: If the device is needed by TCPIP, use the VARY TCPIP command to restart the device.

System Programmer Response: None

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

OE ROUTED Messages

EZZ4820—EZZ4998

This section contains OE ROUTED messages.

EZZ4820I Modify command ignored, incorrect parm(s): *parms*

Explanation: Incorrect parameter(s) were passed to OE RouteD from a MODIFY command.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the parameter(s) from the MODIFY command line.

Source Data Set: EZBRDMAI

Procedure Name: do_modify, parse_parms

EZZ4823I Tracing debug packets *action timestamp*

Explanation: Debug packets tracing is enabled or disabled. The packets are displayed in data format.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: modifydebuglevel

EZZ4824I Modify command ignored, trace levels exceeded

Explanation: An incorrect number of trace levels (-t's) were passed from a MODIFY command.

System Action: OE RoutedD continues.

User or Operator Response: None.

System Programmer Response: Specify a correct number of -t's in the MODIFY command line parameters.

Source Data Set: EZBRDMAI

Procedure Name: parse_parms

EZZ4825E *name***/udp:**

Explanation: A UDP port number for the router service *name* was not assigned in the SERVICES file or data set.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: The following search order is used to find the services data set or file:

1. /etc/services
2. *userid*.ETC.SERVICES for TSO/E or *jobname*.ETC.SERVICES for a batch request
3. *hlq*.ETC.SERVICES

Verify that the SERVICES file or data set has an entry of the form: *name* port/udp. The entry must start in column one and entered in lowercase. Also, verify that port has been reserved for OE RouteD under the PORT statement in the *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4826E Terminating since clients require a socket

Explanation: OE RouteD attempted to open a socket on a well known port (specified in SERVICES file or data set) but the open was not successful, or the socket could not be bound to an internet address and port number. Other routers will not be able to communicate with OE RouteD because a socket is unavailable; therefore, OE RouteD will shut down.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that another program is not using the OE RouteD's port and that the port has been reserved in the PORT statement of *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4827E Usage: orouted ((–t)(–t)(– t)(–t))(–sd| –sv|–sdv|–svd|–st) (–q) (–g)(–dp)(–ep)(–d) (–h)(–hv)

Explanation: OE RouteD was passed parameters that are not supported or were not formatted correctly.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that the parameters are correct and are separated by spaces. See *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4828I Input parameter(s): *parameter*

Explanation: This message displays the input parameters of the OE RouteD server as entered by the user.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: parse_parms

EZZ4829I Waiting for incoming packets

Explanation: OE RouteD is waiting for datagrams to arrive from other routers. Each time that OE RouteD finishes processing an event such as an incoming datagram or a timer that expires, OE RouteD issues this message and waits for the next event. These messages should occur at least every 15 seconds.

System Action: Processing is suspended until a timer expires or a datagram arrives from another router.

User or Operator Response: None.

System Programmer Response: If 6 or more of these messages occur consecutively, OE RouteD is not receiving any datagrams from other routers. Verify that another router is active on a directly con-

nected network. If so, examine the status of the appropriate network interface.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4830I The main select was interrupted

Explanation: An error occurred while OE RouteD was waiting for an event to occur. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Verify that TCPIP is running.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4831I Send delayed dynamic update

Explanation: A routing update that was delayed to prevent packet storms has been transmitted. This occurs 2–5 seconds after a dynamic update has been issued.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4832I While receiving a packet

Explanation: An error occurred while attempting to receive a packet from a client. The incoming packet is discarded. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Refer to the next error message and correct the error.

Source Data Set: EZBRDMAI

Procedure Name: process

EZZ4833E A socket could not be created

Explanation: OE RouteD could not open a new socket. A subsequent message will be issued with more error information.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that TCPIP is active, that another program is not using the OE RouteD's port, and that the port has been reserved in the PORT statement of *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4834E Broadcasting cannot be enabled on the socket

Explanation: OE RouteD cannot enable the socket for broadcasting. OE RouteD must be able to broadcast over interfaces that support broadcasting to communicate with other routers.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4835E The socket bind failed

Explanation: OE RouteD was unable to associate an internet address and port number with the newly created socket. This might be because another application is using the port.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that no other application is using the router port. Use the onetstat -a command and look for a port in the Local Socket column that matches the router port in the SERVICES file or data set. You should reserve this port for OE RouteD's exclusive use by adding an entry to the PORT statement in the *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4836I ***function*****: socket create failed**

Explanation: OE RouteD was unable to obtain a socket for temporary use by *function* while initializing interfaces or obtaining interface information from TCPIP's interface tables. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: various

Procedure Name: *function*

EZZ4837I ioctl (get interface configuration)

Explanation: OE RouteD encountered an error while attempting to obtain the network interface configuration. A subsequent message will be issued with more error information. The most likely error is ENOBUFS, which indicates an internal TCPIP problem. The interface in error is skipped.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4838I *interface*: **ioctl (get interface flags)**

Explanation: OE RouteD encountered an error while obtaining the interface flags for *interface*. A more specific message follows, which indicates the error returned from the ioctl. The interface is skipped.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: If the error is ENOENT, verify that a BSDROUTINGPARMS statement is in the *hlq*.PROFILE.TCPIP data set. Otherwise, contact your IBM Software Support Center. For more information, see *OS/390 TCP/IP OpenEdition Configuration Guide*.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4839I *interface*: **ioctl (get destination address)**

Explanation: OE RouteD encountered an error while attempting to obtain the destination address from the point-to-point interface *interface*. A subsequent message will be issued with more error information. The interface in error is skipped.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4840I *interface*: **ioctl (get broadcast address)**

Explanation: OE RouteD encountered an error while attempting to obtain the broadcast address for interface *interface*. A subsequent message will be issued with more error information. The interface is skipped.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4841I *interface*: **ioctl (get metric)**

Explanation: OE RouteD encountered an error while obtaining the metric for interface *interface*. A subsequent message will be issued with more error information. The metric for the interface is set to zero.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4842I Ignoring incorrect metric on interface *interface*

Explanation: An incorrect metric value was specified for this interface in the BSDROUTINGPARMS statement. The default metric of 0 (zero) is used.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Specify a correct metric value in the range from 0 to 14.

Source Data Set: EZBRDINI

Procedure Name:

EZZ4843I *interface*: **ioctl (get netmask)**

Explanation: OE RouteD encountered an error while obtaining the subnet mask for *interface*. A subsequent message will be issued with more error information. The interface in error is skipped.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4844I ifinit: out of memory

Explanation: OE RouteD is unable to allocate the free storage needed to process an interface.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Increase the region size and restart OE RouteD.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4845I *entry*

Explanation: A warning is about to be issued for the indicated entry in the GATEWAYS file or data set. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4846I *entry*

Explanation: An error is about to be issued for the indicated entry in the GATEWAYS file or data set. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4847I Second element ignored, changing to 'active'

Explanation: The second element in the active gateway entry is detected to be not valid. This suggests that a configuration error might have occurred. OE RouteD will assume the gateway entry to be an active gateway. The second element is changed to active.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4848I Gateway type '*type*' incorrect for active gateway

Explanation: An active GATEWAYS entry has a route type other than "active". GATEWAYS entries classified as active must have a route type of "active". The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4849I OE RouteD Server started

Explanation: OE RouteD has completed initialization of network interfaces with direct routes and is ready to process RIP packets for IP dynamic routing. Any routes defined in the GATEWAYS file or data set have also been processed.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4850I Processing interface *interface*

Explanation: The indicated interface was found in TCPIP's interface table and is being added to OE RouteD's interface table.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4851I First two elements must be 'active' for active gateway

Explanation: A GATEWAYS entry for the gateway definition has a route type of "active", but the first 2 elements are not defined as active, as required for an active gateway definition. The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4852I Incorrect gateway type '*type*'

Explanation: The GATEWAYS entry for the gateway definition has a gateway type that is not valid. Valid gateway types are: active, external, or passive. The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4853I packet from unsupported address family *family***, cmd** *command***.**

Explanation: An RIP packet was received by OE RouteD from a foreign router in an unsupported address family. Only the internet address family is supported. The packet is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Ignore the error or locate the foreign router that originated the unsupported RIP packet and have it stopped.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4854I RIP version 0 packet received from *router***! (cmd** *command***).**

Explanation: An RIP Version 0 packet was received from the specified router. This version of RIP is obsolete and not supported by OE RouteD. The Version 0 packet is discarded.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the router that is sending the version 0 packets.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4855I trace command from unknown router *router*

Explanation: A TRACEON or TRACEOFF packet was received from a router that is either not directly attached, is reached using a passive interface, or reached using an interface that cannot handle broadcast or point-to-point traffic. The packet is discarded.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: If the originating router directed the packet using a passive link, an indirect path, or an interface that cannot handle broadcast or point-to-point traffic, then correct the originating router. Otherwise contact your IBM Software Support Center.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4856I Incorrect packet from passive interface *interface*

Explanation: OE RouteD has received a response packet from a local passive interface *interface*. Although OE RouteD receives routing updates from a client for passive interfaces, it cannot produce the routing updates. The packet is discarded.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate the router that is producing the incorrect response packet and correct the problem.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4857I Packet from unknown router *ipaddr* **(***reason***)**

Explanation: A Routing Information Protocol (RIP) response was received from a router that is not directly connected by a broadcast network, point-to-point (NCST) network, or an active gateway as defined in GATEWAYS file or data set. The *reason* is one of:

Reason	Explanation
Interface in strange state	The network does not support broadcast or point-to-point transmissions.
Iflookup failed	Not directly connected.
this Link is PASSIVE!	Cannot update.

The packet is discarded.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the router that originated the packet.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4858I route from *router* **in unsupported address family** *family*

Explanation: An incoming route from another router is in an address family that is not supported by OE RouteD. Only internet addresses are supported. The route is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the router that originated the route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4859I bad host *host* **in route from** *router*

Explanation: A prohibited internet address *host* was received in an update from router *router*. A previous message indicates the nature of the problem with the address. The route is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the *router* that originated this route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4860I bad metric *metric* **in route to** *destination* **from** *router***.**

Explanation: A route was received from *router* that contained a metric that was not in the range 1–16. The route is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the *router* that originated this route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4861I Send dynamic update

Explanation: Changes to the routing tables have occurred and enough time has passed since the last update so OE RouteD is allowed to update adjacent routers immediately rather than waiting until the next broadcast timer expires. Dynamic updates are not made if they occur too close to another update because network performance might be affected, and they are not made if another dynamic update is pending.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4862I Delay dynamic update

Explanation: Changes have been made to OE RouteD's routing tables, but a dynamic update cannot be made because another update occurred recently, or another dynamic update is already pending. When the last update was made, a timer was set for 2–5 seconds from that time. When this timer expires, all pending dynamic updates are sent.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4863I Inhibit dynamic update for *number* **seconds**

Explanation: A dynamic update has been sent, further dynamic updates are not sent during the next 2–5 seconds.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4864I inet_output: sendto destination *destination*

Explanation: An error occurred while attempting to transmit a packet. The packet is not sent. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDAF

Procedure Name: inet_sendto

EZZ4865I Trace buffers not initialized for interface *interface*

Explanation: One or both of the trace buffers for the specified interface could not be initialized. Tracing is limited.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Increase the region size for OE RouteD and restart.

Source Data Set: EZBRDTRC

Procedure Name: traceinit

EZZ4866I Tracing *action timestamp*

Explanation: Tracing is enabled or disabled.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: various

EZZ4867I Displaying internal *type* **table**

Explanation: OE RouteD's internal IP routing or interface table is displayed for diagnosis.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT, EZBRDIF

Procedure Name: dsp_rrtables, dsp_iftables

EZZ4868I Tracing actions *action timestamp*

Explanation: The current tracing level of actions has been either enabled or disabled. If enabled, messages will be issued for actions such as adding, changing, or deleting a route. Additional messages for actions such as waiting for incoming packets and dynamic updates are also issued.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4869I Tracing packets *action timestamp*

Explanation: The current tracing level of packets has either been enabled or disabled. If enabled, messages will be issued for packets sent and received, in addition to the output displayed for the actions level.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4870I Tracing history *action timestamp*

Explanation: The current tracing level of history has either been enabled or disabled. If enabled, messages will be issued for history tracing data on a per-interface basis, in addition to output displayed for the packets level. The history tracing data is displayed whenever an interface becomes inactive. It shows the latest traces of actions, packets, and packet contents before the interface became inactive.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4871I Tracing packet contents *action timestamp*

Explanation: The current tracing level of packet contents has either been enabled or disabled. If enabled, messages will be issued which displays the contents of packets sent or received, in addition to output displayed at lower tracing levels. Additional messages such as request for full routing tables and unknown address family in routing information are also issued.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4882I *timestamp***:**

Explanation: A full time stamp is issued showing date and time so that traces that exceed one calendar day can be interpreted correctly.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: traceaction

EZZ4883I *action* **destination** *destination***, router** *router***, metric** *metric* **flags** *flags* **state** *state* **timer** *timer*

Explanation: A route to *destination* is added, changed, or deleted depending on the value of *action*. The following values for *action* are allowed:

Value	Description
ADD:	A route to *destination* is added through *router* at a metric of *metric*.
CHANGE FROM:	The route to *destination* is changed, the old values are displayed.
CHANGE TO:	The route to *destination* is changed, the new values are displayed.
DELETE:	The route to *destination* is deleted.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: trace

EZZ4884I rtadd: incorrect address family

Explanation: An attempt is being made to add a route which is to a destination in an unsupported address family. Currently, OE RouteD only supports routes to Internet addresses.

System Action: OE RouteD continues, the route is not added.

User or Operator Response: None.

System Programmer Response: Identify the router which generated this route and correct the problem.

Source Data Set: EZBRDRMT

Procedure Name: rtadd

EZZ4885I rtadd: out of memory

Explanation: OE RouteD does not have enough storage to allocate a new route table entry to add a new route.

System Action: OE RouteD continues, the route is not added.

User or Operator Response: None.

System Programmer Response: Increase the region size for OE RouteD. Each route table entry requires 64 bytes.

Source Data Set: EZBRDRMT

Procedure Name: rtadd

EZZ4888I Interface *interface* **is passive**

Explanation: The *interface* is in a passive state, meaning that RIP traffic is disabled for the interface. Routing updates will not be broadcast to the interface and incoming routing updates are ignored. This may be the result of a RIP I/O filter option specified in the GATEWAYS file or data set. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4889I CHANGE metric destination *destination***, router** *router***, from** *old metric* **to** *new metric*

Explanation: The metric for the route to *destination* is changed from *old metric* to *new metric*. OE RouteD will change the metric if it has received a RIP packet from router *router* thas has a different metric than the one in OE RouteD's tables. The message is also issued when a route times out, and OE RouteD changes the metric value to 16 (unreachable).

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: tracenewmetric

EZZ4890I toall: requested to skip interface *interface*

Explanation: The interface *interface* is skipped because the interface has already received notification of a routing change. If a broadcast of the routing table has not been sent recently and a routing change has occurred, a dynamic routing update will be broadcast to other interfaces to inform adjacent routers of the routing change.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4891I * Packet history for interface** *interface* *******

Explanation: Tracing is set at the history level and the history trace data for the inactive interface is displayed.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumpif

EZZ4892I * End packet history *****

Explanation: Tracing is set at the packet history level and this message ends the history trace data for the inactivated message.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumpif

EZZ4897I *direction***: no packets.**

Explanation: Either the input or output trace buffer, depending on the value of *direction*, is empty.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumptrace

EZZ4898I *direction* **trace:**

Explanation: Tracing is set at the history level and either the input or output trace buffer is about to be displayed, depending on the value of *direction*.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumptrace

EZZ4899I *RIPcmd direction router* **->** *port*

Explanation: An RIP datagram has been received from or is about to be transmitted to another router, depending on the value of *direction*. A value of *to* indicates an outbound datagram, while a value of *from* indicates an inbound datagram. The type of the datagram is determined by the value of *RIPcmd* and one of the following:

*RIP*cmd	**Description**
REQUEST	Indicates that the routing table information is requested. Individual routes or a complete routing table may be requested.
RESPONSE	Indicates that the routing table information is broadcast.
TRACEON	Request to start tracing to a specified data set.
TRACEOFF	Request to stop tracing and close the current tracing data set.

The value of *port* is either the port number on which the datagram came in or goes out on, or in the case of 0, it indicates that an outbound datagram will go out on the port assigned to *router* in the SERVICES data set.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4900I Bad cmd *hex direction router* **->** *port timestamp*

Explanation: A malformed packet was encountered during tracing. The source of the packet is either *router* if *direction* is *from*, or OE RouteD if *direction* is *to*.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: If the source of the packet is *router*, correct this router. Otherwise, contact your IBM Software Support Center.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4901I (truncated record, len *len***)**

Explanation: An RIP RESPONSE datagram was received that did not end on a route boundary. Either the packet extends beyond the last route, or it has been truncated.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Verify that the incoming packet was built correctly and can be processed by other routers. Contact your IBM Software Support Center if the problem appears to be with OE RouteD.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4902I destination *destination* **metric** *metric*

Explanation: A route to *destination* with the indicated metric is displayed by another router or OE RouteD, depending on the contents of the last EZZ4899I message. A route is never advertised over the interface from which it was received.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4903I request for full tables

Explanation: A request for a complete routing table has been sent or received depending on the contents of the last EZZ4899I message. This message is sent by OE RouteD at startup over each interface so that OE RouteD can obtain routing tables from all adjacent routers.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4904I Unknown address family (*family***) metric** *metric*

Explanation: A route was found in the trace with an unknown address family.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: If the route originated from another router, locate that router and correct it. If the route originated from OE RouteD, contact your IBM Software Support Center.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4905I TraceOn file = *dataset*

Explanation: A TRACEON packet was received, trace data is requested to go to *data set*.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4909I Error adding route to *destination* **through router** *router***, metric** *metric*

Explanation: An error occurred while attempting to add a route to *destination*. A subsequent message will be issued with more error information.

System Action: OE RouteD continues, the route is not added.

User or Operator Response: None.

System Programmer Response: See EZZ4910E.

Source Data Set: EZBRDRMT

Procedure Name: tcp_add

EZZ4910I *function*: **SIOCADDRT ioctl failed**

Explanation: OE RouteD was unsuccessful in adding a new route. The error occurred in the indicated function and is one of the following:

Function	Description
rtadd	OE RouteD attempted to add a new route to the TCPIP IP routing table.
rtchange	OE RouteD attempted change the metric of a route in TCPIP's IP routing table by deleting the old route and adding a new one with the updated metric.

A subsequent message will be issued with more error information. Most likely the error is the result of a bad GATEWAYS entry, and common errors are:

Error	Description
EACESS	OE RouteD is not authorized to add routes to TCPIP's IP routing table. Permission is granted to OE RouteD in managing the TCPIP's IP routing table if OE RouteD's catalogued procedure name is specified in the obey list.
EEXIST	The route already exists in TCPIP's tables either because of a GATEWAY statement, a previous job execution of OE RouteD, or lack of success in deleting the route.
ENETUNREACH	TCPIP does not have a route to the router (nexthop) for this route. Routes in the GATEWAYS file or data set must have reachable routers.
ENOBUFS	TCPIP did not have a free IP Route Control Block to add the route to the routing table.
ENOENT	TCPIP did not have a link interface defined for the route to be added to.
EINVAL	An incorrect parameter has been passed to TCPIP.

System Action: The route is not added. OE RouteD continues.

User or Operator Response: None.

System Programmer Response: These errors require the following responses:

Error	Response
EACESS	Make sure that the name of the catalogued procedure is specified in the obey list (OBEY statement) in the *hlq*.PROFILE.TCPIP. Ensure that the name is spelled correctly.
EEXIST:	Clear the IP routing table before restarting OE RouteD. An obeyfile with the word *GATEWAY* removes all routes.
ENOBUFS:	Increase the IPROUTEPOOLSIZE in *hlq*.PROFILE.TCPIP and restart TCPIP.
ENETUNREACH:	Correct the GATEWAYS file or data set, either add a route to the router, or select another router for this route.
ENOENT:	Define the link interface on the BSDROUTINGPARMS statement in the *hlq*.PROFILE.TCPIP. An obeyfile with the BSDROUTINGPARMS definition(s) may be used to add the link interface(s) to TCPIP.
EINVAL:	Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: tcp_add

EZZ4912E Error deleting route to *destination* **through router** *router*, **metric** *metric*

Explanation: An error occurred while attempting to delete a route to *destination*. A subsequent message will be issued with more error information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Refer to the explanations for the specific error messages displayed with this one.

Source Data Set: EZBRDRMT

Procedure Name: tcp_del

EZZ4916I *action* **route to interface** *interface* **(timed out)**

Explanation: The route to *interface* is changed or deleted depending on the value of *action*. The value *changing* indicates that *interface* will be deleted then readded. The value *deleting* indicates that the route to *interface*'s network or subnet will be deleted. The interface remains in OE RouteD's interface list even though the route is deleted.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT

Procedure Name: rtchange

EZZ4918I *function*: **SIOCDELRT ioctl failed**

Explanation: OE RouteD was unsuccessful in deleting a route. The error occurred in *function*, and is one of the following:

Function	Description
rtdelete	OE RouteD attempted to delete a route from TCPIP's routing table.
rtchange	OE RouteD needed to change a route, but TCPIP does not allow routes to be changed explicitly. Instead, OE RouteD attempted to delete the old route so that a new route can be added with the updated values.
rtdeleteall	OE RouteD attempted to delete all routes from TCPIP's routing IP routing table.

A subsequent message will be issued with more error information. The most likely error is ESRCH, which indicates that the route that is to be deleted does not exist in TCPIP's routing tables.

System Action: OE RouteD continues, the route is not deleted.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: tcp_del

EZZ4919I Using backup interface *interface*

Explanation: A primary interface became inactive and a secondary *interface* is serving as the backup.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: delrouteforif

EZZ4920I Add *type* **route ignored, route exists on interface** *interface*

Explanation: A network, subnetwork, or destination type route already exist on *interface* that is active. As a general rule, no more than one route can be defined to a single destination and a route is added to the first available interface in the (sub)network according to the order in the HOME statement of *hlq*.PROFILE.TCPIP.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4921I Deleting route to interface *interface***? (timed out?)**

Explanation: An interface route is deleted even though the route exists in TCPIP's tables. One possible cause is that the route was deleted due to a timed out situation. Another reason for the deletion is defining a passive route in the GATEWAYS file or data set, when the route was also defined in the BSDROUTINGPARMS or HOME section.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: rtdelete

EZZ4922I Option(s): *options*

Explanation: Additional OE RouteD options, specified in a GATEWAYS file or data set, are being processed. See *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4923E Incorrect option: *option*

Explanation: In the line entry for the GATEWAYS file or data set, the options definition has an incorrect option. Although other options may be processed normally, the incorrect option is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4924I Re-installing interface *interface*

Explanation: The previously deleted interface *interface* is reinstalled, because traffic has been detected over this interface.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4925I Start of GATEWAYS processing:

Explanation: The GATEWAYS file or data set member is about to be processed. Processing messages may follow for the entries in the file or data set.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4926I End of GATEWAYS processing

Explanation: Processing is completed for the GATEWAYS file or data set.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4927I Zero metric not allowed, changing to one

Explanation: In the line entry for the GATEWAYS file or data set, the metric has a value of 0, which is not allowed. The range of valid metric values is 1 through 15. The metric is changed to a default value of 1.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4928E Out of memory while processing GATEWAYS

Explanation: OE RouteD did not have enough storage available to process the GATEWAYS file or data set, which contains the addresses of gateways on the network.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: Increase the region size for OE RouteD and restart OE RouteD. If the problem persists, contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4929I Port *port* **assigned to** *name*

Explanation: OE RouteD listens for traffic from other routers on the specified port number *port* assigned to router service *name*.

System Action: TCPIP continues. OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4932I ******************************

Explanation: Two of these banners enclose a message that might need attention when viewing the output.

System Action: TCPIP continues. OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Read the enclosed message and, when necessary, resolve the indicated error.

Source Data Set: various

Procedure Name: various

EZZ4933I Unable to open a GATEWAYS *file/dataset* **(***filename***)**

Explanation: A gateways file or data set was not specified, or could not be opened. The indicated file or data set is the file or data set that OE RouteD attempted to open.

System Action: TCPIP continues. OE RouteD continues.

User or Operator Response: None.

System Programmer Response: The OE RouteD server uses the following search order to locate the GATEWAYS configuration data set or file:

1. If the environment variable GATEWAYS_FILE has been defined, .OE RouteD uses this value as the name of an MVS data set or HFS file to access the gateways file. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/gateways
3. *hlq*.ETC.GATEWAYS

   Note: Only the first file in the search order that can be opened will be read to determine the gateway statements.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4934I Opening GATEWAYS *file/dataset* **(***filename***)**

Explanation: The GATEWAYS file or data set is being opened and the entries in the file or data set are read in for input.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4935E Incorrect parameter: *parameter*

Explanation: An incorrect parameter was passed to OE RouteD. The parameter could either be passed from the command line parameters or from the default parameter list in the start procedure JCL.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the parameter from the command line parameters or in the default parameter list of the start procedure JCL.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4936I Adding *type route* **route** *destination* **via gateway** *gateway***, metric** *metric*

Explanation: The indicated route, defined in the GATEWAYS file or data set, is being added to OE RouteD's routing table. The route to the gateway will not be replaced by a competing RIP route.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4937I Subnetwork mask unknown for *destination***, using network route** *route*

Explanation: The indicated subnetwork route, defined in the GATEWAYS file or data set, was explicitly coded as a “net” route type. Because the subnetwork mask for the destination subnetwork is unknown, OE RouteD replaces the subnetwork route with a network route. OE RouteD currently does not support variable subnetting.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT

Procedure Name: rtadd

EZZ4938I Adding active gateway *ip_addr***, metric** *metric*

Explanation: An active gateway *ip_addr* with metric *metric*, defined in the GATEWAYS file or data set, is being added to OE RouteD's routing table. The route to the active gateway will be treated as a network interface.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4939E Trace levels exceeded maximum of 4 -t's

Explanation: The maximum number of -t's that are part of a valid request is 4 (-t-t-t-t). Each -t indicates an additional level of tracing.

System Action: OE RouteD ends abnormally.

User or Operator Response: Specify a valid -t parameter for the trace request you would like. *OS/390 TCP/IP OpenEdition Configuration Guide* describes the OE RouteD parameters.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4940I Point-to-point interface, using dstaddr

Explanation: The new interface is point-to-point, such as a SNALINK interface. An implicit route is being added based on the interface definition. The dstaddr field coded in the TCPIP profile will be used as the route destination rather than a network or subnetwork route.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4941I Point-to-point interface, using (sub)network address

Explanation: The new interface is point-to-point, such as a SNALINK interface. An implicit route is being added based on the interface definition, and the network or subnetwork route will be used as the route destination.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4942I Not an internal interface

Explanation: The new interface does not appear to be associated with a real device. Most likely the interface is a pseudo-interface, created as a result of an active gateway definition in the GATEWAYS file or data set.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4943I Adding *type* route for interface

Explanation: The route using the network, subnetwork, or host type is being added to the interface in OE RouteD's routing table.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4944I Function *function*: address family is out of range. The address is *address*.

Explanation: The route to the destination address has an unsupported address family. OE RouteD cannot determine the network based on the address, therefore it cannot determine an interface that serves the logical network.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Non-IP addresses are not supported in this version of OE RouteD. Either ignore the message, or have the router that originated the route stop sending non-IP addresses to this router. Look back through the output to find the last ADD or CHANGE for this destination to obtain the IP address of the router.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4945I ifwithnet: compare with *value*

Explanation: The route's network address is being compared to one of OE RouteD's interface entries for a network number match.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4946I ifwithnet: interface has bad address family

Explanation: One of OE RouteD's interface entries address families is not valid. OE RouteD cannot determine the network based on the address, and cannot determine an interface that serves the logical network. Valid address family is AF_INET for the internet domain.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Either ignore the message, or correct the interface with a correct address family. Change the address family to a valid value. If the socket address is in an internet addressing family, the addressing family can be defined in the IN.H header file. See *OS/390 TCP/IP OpenEdition Programmer's Reference* for information about valid addressing families. If the interface cannot be corrected, contact your IBM Software Support Center.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4947I netmatch *ipaddr1* **and** *ipaddr2*

Explanation: A network number match was found for the first address in the message with one of OE RouteD's interface entries that has the second address as its network address.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4948I This interface is not point-to-point

Explanation: The interface OE RouteD is currently handling is not a point-to-point interface. OE RouteD will create a route to the network for the interface.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: if_ifwithnet

EZZ4949I Interface *interface* **not up**

Explanation: OE RouteD determined that the indicated interface is not active. No routes will be added unless the interface is activated or reactivated.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT, EZBRDINI

Procedure Name: toall, ifinit

EZZ4950I Interface *interface* **ignored**

Explanation: The indicated interface address family is not valid. The interface is ignored to prevent network or subnetwork routes from being added.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Either ignore the message, or correct the interface with a valid address family. A common cause of an interface with an incorrect address family is that the interface is undefined in HOME statement of *hlq*.PROFILE.TCPIP. This may be normal in the case where multiple interfaces are reserved for Offload processing but are not defined as primary interfaces (for example, the PRIMARYINTERFACE statement in *hlq*.PROFILE.TCPIP). If the interface cannot be corrected, contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4954I *timer_value* **minute timer expired for route to** *destination*

Explanation: No Routing Information Protocol (RIP) packets have been received from OE RouteD for the route to the destination *destination* in the last *timer_value* minutes. It is assumed that the destination router is no longer active. Depending upon the *timer_value*, one of the following actions is taken:

Value	Action
3	The route will have its metric changed to infinity for the next 2 minutes. The metric change is necessary to alert adjacent routers that the route is to this destination is unreachable. If OE RouteD receives any RIP packets from the destination router during the 2 minute interval, OE RouteD will restore the route by changing the metric to a valid one based upon the received RIP packet.
5	The route will be deleted from OE RouteD's routing table.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response:

System Programmer Response: These actions require the following responses:

Value	Response
3	If the route was broadcast for a while, and then suddenly stopped, look for an adapter problem, or a problem with the physical line.
5	Examine OE RouteD trace output and determine when the broadcasting has stopped for the route to the destination router. If the route to *destination* was only broadcast once in response to OE RouteD's request for full route tables, then the problem may be with the way RIP packets are broadcast.

In these cases, determine if OE RouteD is receiving the RIP packets by obtaining a TCPIP trace with MORETRACE IPUP and look for "discarding broadcast packet" messages.

If the route was being broadcasted for a while, and then suddenly stopped, look for an adapter problem, or a problem with the physical line. A MORETRACE IPUP can confirm the lack of traffic for the route's interface. Contact your IBM Software Support Center for assistance.

Source Data Set: EZBRDTMR

Procedure Name: timer

EZZ4955I Ignoring route *destination***, filtered out**

Explanation: The specified destination route is being filtered out from the routing information in a outgoing RIP broadcast or a received RIP update. The destination route matches one of the RIP input and output filters, defined in OE RouteD gateways data set or file, and is being selected for removal from the routing information.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP, EZBRDOUT

Procedure Name: rip_input, supply

EZZ4956I Deleting interface *interface_name*

Explanation: OE RouteD is deleting the interface from its interface tables. The interface is unusable, since it may have been dynamically deleted via the VARY TCPIP command for a new HOME list. See OS/390 TCP/IP OpenEdition Configuration Guide for more information about the VARY TCPIP command.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: ifdelete

EZZ4957I *timer_value* **second timer expired (***reason***)**

Explanation: Depending upon the *timer_value* and *reason*, one of the following actions is taken:

Value	Reason	Action
30	broadcast	Every 30 seconds, a timer expires that indicates that OE RouteD must broadcast its routing tables to adjacent routers. At this time, OE RouteD will send RIP response packets for each of OE RouteD's interfaces that allow broadcasting to the adjacent routers.
m	rescan kernel for interfaces	Every *m* seconds, a timer expires that indicates that OE RouteD must rescan TCPIP's interface table (kernel) for any activated, reactivated, or deactivated interfaces. At this time, OE RouteD will add any new interfaces to its interface table, recognize interface order according to HOME and PRIMARYINTERFACE statements in *hlq*.PROFILE.TCPIP, or marks the interface as active or inactive. In addition, OE RouteD will add routes for the activated or reactivated interfaces and if necessary, issue alerts (triggered updates) to adjacent routers for deactivated interfaces to attempt routing recovery.
n	poll interfaces for status	Every *n* seconds, a timer expires that indicates that OE RouteD must poll TCPIP's interface table for any activated, reactivated, or deactivated interfaces. OE RouteD will mark the interface as active or inactive and if necessary, issue alerts (triggered updates) to adjacent routers for deactivated interfaces to attempt routing recovery.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTMR

Procedure Name: timer

EZZ4958I supply *destination* **->** *port* **via** *interface*

Explanation: OE RouteD is broadcasting a routing information protocol (RIP) datagram to the destination address. The destination address can be either a broadcast address or a host address. If the port is zero, the current OE RouteD port will be used from the SERVICES file or data set. Otherwise, the datagram will be transmitted to the specified port over the interface to the destination.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: supply

EZZ4959I Interface *interface* **up**

Explanation: The specified interface, which was not previously active, is now active. The interface might have been inactive or re-installed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4960I Incorrect host or (sub)network address '*destination***'**

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition has an incorrect destination address. The destination must be either a resolvable host name, or an IP address in dotted decimal notation. The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4961I Incorrect gateway address '*gateway***'**

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition has a gateway address that is not valid. The gateway address must be either a resolvable gateway name, or an IP address in dotted decimal notation. The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4962I Incorrect route type '*type***'**

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition route type is not valid. Allowable route types are host, for host route, net, for network or subnetwork route, and active for a route to be treated as a network interface. The GATEWAYS entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4963I (no gateway definitions)

Explanation: The GATEWAYS file or data set does not contain any definitions for configuring the routing tables on your host.

System Action: TCPIP continues.

User or Operator Response: Include appropriate definitions in the GATEWAYS file or data set for configuring the routing tables on your host before starting OE RouteD.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4964I Incorrect metric, changing to one

Explanation: In the line entry for the GATEWAYS file or data set, the metric has an incorrect value. The metric is changed to 1 because the gateway internet address matches one of the home internet addresses on the originating host. The destination host or network address is one hop away.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4965I Virtual interface

Explanation: The new interface is a virtual and is used for fault tolerance support. The virtual interface always stays active and will never see a physical failure. Network and/or subnetwork virtual routes are being added to the interface.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4972I Incorrect *type* **value:** *value*

Explanation: In the line entry for the GATEWAYS file or data set, the options definition contains an incorrect value. Although other options may be processed normally, the incorrect option is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4973I Unknown interface name (*interface***) and/or address (***ip_addr***)**

Explanation: In the line entry for the GATEWAYS file or data set, the options definition contains interface information that is not in OE RouteD's interface table. Most likely, the interface name is misspelled or the interface's internet address is specified incorrectly. Although other options may be processed normally, the incorrect option is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4974I Issuing Kill Route Messages

Explanation: OE RouteD has detected an outage on a primary or alternate adapter and is issuing an alert 'Kill Route Messages' to adjacent routers over another interface providing the redundant path. This alert is used to notify adjacent routers to remove their routes to the failing adapter so that they can receive new routing information from the new interface and perform routing switches immediately.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4975I NLSPATH environment variable has more than one instance of '%L'. Using OE RouteD's default NLSPATH: *path*

Explanation:

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: If the default NLSPATH *path* is acceptable, ensure that OE RouteD's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so it has only one instance of '%L', ensure that OE RouteD's catalogs are located in that path, and stop and restart OE RouteD.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4976I NLSPATH environment variable does not end with '%N'. Using OE RouteD's default NLSPATH: *path*

Explanation: The NLSPATH environment variable has incorrect syntax for OE RouteD's use. '%N' must appear at the end of the string. OE RouteD will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default NLSPATH *path* is acceptable, ensure that OE RouteD's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so it has a '%N' at the end of the string, ensure that OE RouteD's catalogs are located in that path, and stop and restart OE RouteD.

Source Data Set: EZBRDMAI

Procedure Name: validate_nlspath

EZZ4977I NLSPATH environment variable has '%N' that is not at the end of the string. Using OE RouteD's default NLSPATH: *path*

Explanation: The NLSPATH environment variable has incorrect syntax for OE RouteD's use. '%N' must appear at the end of the string. OE RouteD will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default NLSPATH *path* is acceptable, ensure that OE RouteD's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so '%N' appears only at the end of the string, ensure that OE RouteD's catalogs are located in that path, and stop and restart OE RouteD.

Source Data Set: EZBRDMAI

Procedure Name: validate_nlspath

EZZ4978I NLSPATH environment variable is too long to be used by OE RouteD. The maximum length for OE RouteD's use is *max***. Using OE RouteD's default NLSPATH:** *path***.**

Explanation: OE RouteD is unable to use the current setting of the NLSPATH environment variable because its length is greater than *max*. OE RouteD will use *path* to locate its message and reply catalogs.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default NLSPATH *path* is acceptable, ensure that OE RouteD's catalogs are located there. If it is not acceptable, reset the NLSPATH environment variable so its length is less than *max*, ensure that OE RouteD's catalogs are located in that path, and stop and restart OE RouteD.

Source Data Set: EZBRDMAI

Procedure Name: validate_nlspath

EZZ4979I LANG environment variable is longer than *max***. Using OE RouteD's default LANG value:** *lang*

Explanation:

Explanation: OE RouteD is unable to use the current setting of the LANG environment variable because it's length is greater than *max*. OE RouteD will use *lang* to locate its message and reply catalogs within the NLSPATH.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the default value *lang* is acceptable, ensure that OE RouteD's catalogs are located there. If it is not acceptable, reset the LANG environment variable so its length is less than *max*, ensure that OE RouteD's catalogs are located in that path, and stop and restart OE RouteD.

System Action: Processing continues.

Source Data Set: EZBRDMAI

Procedure Name: validate_lang

EZZ4980I Using catalog '*file***' for OE RouteD messages.**

Explanation: The messages issued by OE RouteD will be retrieved from the message catalog named *file*

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4981I Unable to open message catalog '*file***'.**

Explanation: An attempt was made to open OE RouteD's message catalog (named '*file*') in the directory determined by the NLSPATH and LANG environment variables, but the catalog could not be opened A subsequent message will be issued with more error information.

System Action: Processing continues. Default messages will be used.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external catalog is desired, stop the server, correct the problem as indicated by the error text, restart the server. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the IBM C/C++ for MVS/ESA Library Reference Volume 1 for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the OpenEdition MVS Advanced Application Programming Tools publication. If the default messages are acceptable, no action is necessary.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4982I Using OE RouteD's default messages

Explanation: Because OE RouteD was unable to open the message catalog, the default messages will be used.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4983I *errnum* **(ERRNO=***nnnn* **ERRNO2=***nnnnnnnn***)**

Explanation: After a system error, *errnum*, ERRNO and ERRNO2 are displayed

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: sys_err

EZZ4984I Cannot determine the High Level Qualifier (HLQ). Using default HLQ of *hlq*

Explanation: Because OE RouteD was unable to determine the HLQ, the default HLQ of *hlq* will used.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Check the OMVS resolver configuration file or data set to ensure that value of system variable DATASETNAME is set properly.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4985I Setting High Level Qualifier (HLQ) to '*hlq***'**

Explanation: OE RouteD has set the HLQ variable to the system variable DATASETNAME.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4986I OE RouteD could not determine TCPIPjobname, using default of 'INET'

Explanation: OE RouteD could not determine the jobname for the TCPIP stack that it is to associate itself with. A default value of 'INET' will be used for TCPIPjobname.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for OE RouteD to communicate with particular stack the TCPIPjobname should be set in the appropriate resolver configuration file or data set. The following is the search order used to locate the resolver configuration data set or file:

1. If the environment variable RESOLVER_CONFIG has been defined, the resolver uses the value of this environment variable as the name of an MVS data set or HFS file to access the resolver configuration data. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/resolv.conf
3. Any MVS data set that is pre-allocated to a DD-name of SYSTCPD. The use of this technique is discouraged because of restrictions for DD-name allocations during fork() processing.
4. *userid*.TCPIP.DATA for TSO/E or *jobname*.TCPIP.DATA for a batch request
5. SYS1.TCPPARMS(TCPDATA)

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4987I OE RouteD could not establish affinity with '*jobname***'**

Explanation: OE RouteD cannot communicate with the TCPIP stack *jobname*. OE RouteD attempted to use the OE Socket call setibmopt() to associate itself with the TCPIP instance *jobname*. This TCPIP name should be the started procedure name (or identifier if the 'S member.identifier' format of the MVS start command was used) of the TCPIP instance to which OE RouteD is to be associated with. A subsequent message will be issued with the *errcode* and *reason* for the setibmopt() failure.

System Action: OE RouteD ends abnormally.

User or Operator Response: None.

System Programmer Response: For OE RouteD to communicate with a particular stack, the jobname (as determined by the system variable TCPIPjobname) must match "xxxxx" where "xxxxx" is set in the BPXPRMxx member used to start OMVS. "xxxxx" is set in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(EZBPFINI).

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4988I OE RouteD established affinity with '*jobname***'**

Explanation: OE RouteD successfully connected to the stack *jobname*.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4989I OE RouteD MODIFY command processed; see syslog output

Explanation: OE RouteD has processed a MODIFY command.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: do_modify

EZZ4990I OE RouteD server initializing. Level *level*

Explanation: OE RouteD server is starting initialization. Internal OE RouteD level number is displayed.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4991I OE RouteD: sigaction() failed for *signal* **:** *reason*

Explanation: OE RouteD encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, OE RouteD will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

SIGABND	handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.
SIGTERM	handler controls cleanup of resources during termination.
SIGPIPE	handler allows OE RouteD to detect when the connection to OE Traceroute has been terminated.

A subsequent message will be issued with more error information.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4992I OE RouteD: Abend detected

Explanation: This message indicates that an internal programming error caused OE RouteD to abnormally terminate.

System Action: OE RouteD ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: abndhand

EZZ4993I OE RouteD: Terminated

Explanation: OE RouteD received the SIGTERM signal

System Action: OE RouteD exits

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: termhand

EZZ4994I Incorrect routing filter mask '*mask***'**

Explanation: In the line entry for OE RouteD gateways file or data set, the gateway definition has an invalid route filter mask. The mask must be either a resolvable mask name, or a 32-bit value in dotted decimal notation. The gateways entry is ignored.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the gateways file or data set.

Source Data Set: EZBRDINI

Procedure Name: ParseOptions

EZZ4995I Interface *interface* skipped, supply is suppressed

Explanation: The broadcasting of routing information is being suppressed for the specified interface. For this reason, the interface is being skipped while OE RouteD is supplying routing information to list of interfaces. An interface supply option, defined in the OE RouteD gateways file or data set, determines whether or not OE RouteD can supply routing information over an interface.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4997I Unsupported RIP command *command* received from *source*

Explanation: OE Routed has received an unsupported RIP command *command* from *source*.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Locate the router that is producing the unsupported RIP command and correct the problem.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4998I Gateway address '*address*' not on a routing interface in the network

Explanation: The indicated route, defined in OE RouteD's gateways dataset, referenced an unknown routing interface based upon the gateway address. The route definition is ignored.

System Action: RouteD continues.

User or Operator Response: None.

System Programmer Response: Correct the gateways dataset. Verify that the gateway address is correct and a valid routing interface is defined.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

SNMP Agent Messages

EZZ6201—EZZ6274

This section contains SNMP Agent messages.

EZZ6201I SNMP agent: Unable to open message catalog *snmpdmsg.cat*: *additional error text*

Explanation: The SNMP agent was unable to open the message catalog *snmpdmsg.cat* in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be *NLSPATH=/usr/lib/nls/msg/%L/%N*.

System Action: The agent will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *IBM C/C++ for MVS/ESA Library Reference Volume 1* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the *OpenEdition MVS Advanced Application Programming Tools*. If the default messages are acceptable, no action is necessary.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6202I Using catalog *catalog file* **for SNMP agent messages**

Explanation: The SNMP agent has located its message catalog file.

System Action: The agent continues to initialize.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6203I Sigaction for *signal handler* **failed :** *specified action: error text (errno/errno2)*

Explanation: The agent issued a sigaction function for the *specified action* which failed. The *error text* will provide more information about the cause of the error. The *errno* and *errno2* are the returned error codes.

In the message text:

signal handler
One of the following sigaction functions: SIGTTOU, SIGTERM, SIGABEND, SIGPIPE, SIGTTIN

System Action: The agent continues to initialize.

User or Operator Response: None.

System Programmer Response: Correct the problem indicated by *error*. Refer to the *C/C++ Library Reference* for further explanation of the socket errors.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6204I SIGTERM received for SNMP daemon which is now shutting down.

Explanation: The agent received a SIGTERM signal.

System Action: The agent terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: pgmstpd

EZZ6205I OE SNMP agent: Could not determine TCPIP jobname using default of 'INET'

Explanation: The OE SNMP agent, osnmpd, could not determine the jobname for the TCPIP stack that it is to associate itself with. A default value of 'INET' will be used for TCPIPjobname.

System Action: OE SNMP agent continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for the OE SNMP agent to communicate with particular stack the TCPIPjobname should be set in the appropriate resolver configuration file or data set. The following is the search order used to locate the resolver configuration data set or file:

1. If the environment variable RESOLVER_CONFIG has been defined, the resolver uses the value of this environment variable as the name of an MVS data set or HFS file to access the resolver configuration data. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/resolv.conf
3. Any MVS data set that is pre-allocated to a DD-name of SYSTCPD. The use of this technique is discouraged because of restrictons for DD-name allocations during fork() processing.
4. *userid*.TCPIP.DATA for TSO/E or *jobname*.TCPIP.DAT for a batch request
5. SYS1.TCPPARMS(TCPDATA)

An SNMP agent must be associated with a single TCPIP instance, since a portion of the MIB objects supported by a TCPIP instance are actually implemented in the agent.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6206I Unable to open *configuration file* **:** *error text*

Explanation: The agent was unable to open the *configuration* file. This *configuration* file can be the community names file or the file for the default MIB variables. Refer to the *error text* for a more specific reason for the failure. The community names file or the MIB variables file can be in one of several files. The initialization process is searching for a community names or MIB variables file in the prescribed order and reporting on its status.

System Action: The agent continues searching for the next file name.

User or Operator Response: This is an informational message indicating that the particular file in the path was not found.

System Programmer Response: A community names or MIB variables file is required for changing defaults. If the indicted file was not found at all, then the configuration of the agent ends and the agent continues with initialization. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for configuring the community names data set.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6207I Line number *number* **in the** *community name* **file is not in the correct format**

Explanation: The agent was unable to interpret the statement in the *community name* file. The statement had fewer than 3 fields, (community name, IP address, and mask) or was not a comment statement that started with a '*' or '#' character.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. If necessary, reIPL the agent. for configuring the community names data set. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for more information.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6208I Line number *number* **in the** *community name* **file has a community name greater than** *number* **characters**

Explanation: The community name at the specified line number was greater than than the indicated number of characters. The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. In order to pick up the changes, reIPL the agent. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for configuring the community names data set.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6209I Line number *number* **in the** *community name* **file has a network address:** *IP address* **which is not in the correct format**

Explanation: The network address at the specified line number is not in dotted notation(xxx.xxx.xxx.xxx). The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for configuring the community names data set.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6210I Line number *number* **in the** *community name* **file has a network mask:** *IP address* **which is not in the correct format**

Explanation: The network mask at the specified line number is not in dotted notation(xxx.xxx.xxx.xxx). The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for configuring the community names data set.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6211I Error reading the *community names* **file at line number:** *number* **error text:** *error text*

Explanation: The agent was attempting to read the specified line number in the "community names file" but had an error described by the *error text*. Refer to the *error text* for a more specific reason for the failure. The error number is also supplied.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: Determine why there was an i/o error and correct it.

Source Data Set: EZASNAAG.C

Procedure Name: snmp_set_mib_install_defaults

EZZ6212I Unable to open *trap destination* **file :** *error text*

Explanation: The agent was unable to open the *trap destination* file. Refer to the *error text* for a more specific reason for the failure. The trap destination file can be in one of several files. The initialization process is searching for a trap names file in the prescribed order and reporting on its progress attempting to open the files.

System Action: The agent continues searching for the next trap destination file.

User or Operator Response: None.

System Programmer Response: This message can be ignored as long as a trap destination file is eventually found. If no trap destina-

tion file is found, then defaults will be used and message EZZ6213I will be issued for this condition.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6213I Using SNMP trap defaults

Explanation: The agent was unable to open any trap destination files.

System Action: The agent uses the default trap destination settings.

User or Operator Response: None.

System Programmer Response: The agent continues initializing. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* for creating a trap destination data set.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6214I Using *trap destination file* **for trap destination file**

Explanation: The agent has found and opened the *trap destination file*

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6215I Line number *number* **in the** *trap destination* **file is not in the correct format**

Explanation: Trap initialization could not process the statement identified by the indicated line number.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the *trap destination* file and correct it. Restart the agent after the statement has been corrected. Refer to the *OS/390 TCP/IP OpenEdition Configuration Guide* on creating a trap destination data set.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6216I line number *number* **in the** *trap destination* **file failed host_lookup for ip address:** *hostname*

Explanation: The host_lookup function could not resolve the *hostname* into an ip address for the given *trap destination* at line number *number* in the *trap destination* file.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the *trap destination* file and correct it. Restart the SNMP agent.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6217I Line number *number* **in the** *trap destination* **file** *protocol* **is not a supported connection**

Explanation: The only trap protocol for trap reporting is UDP. At indicated line number in the trap destination file something other than "UDP" was specified.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement at line number *number* in the *trap destination* file and correct it.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6218I Error reading the *trap destination* **file at line number:** *line number* **- error text:** *error text*

Explanation: There was an error reading the *trap destination* file at the line indicated by *line number*. The *error text* will provide more information about the cause of the error.

System Action: The agent closes the *trap destination* file current statement and continues initialization.

User or Operator Response: None.

System Programmer Response: Refer to the error text to determine what is wrong with the *trap destination* file.

Source Data Set: EZASNACF.C

Procedure Name: snmp_configure_trap_v1

EZZ6219I rc=*rc* **(***error text***) (expect** *rc* **(***additional error text***)) from (***SNMP function name***)**

Explanation: There was an error while the agent attempted to add a trap destination. Refer to the additional information provided in the message to determine the cause. The *SNMP function name* is the failing SNMP function name. The *error text* will provide more information about the cause of the error.

System Action: The agent ignores the problem and continues with initialization.

User or Operator Response: None.

System Programmer Response: Try to correct the problem with the help of the information in this message.

Source Data Set: EZASNACF.C

Procedure Name: add_trap_destination

EZZ6220I Unknown trap version *version*

Explanation: There was an error while the agent attempted to add a trap destination. The *version* was specified and it is not recognized as valid.

System Action: The agent ignores the problem and continues with initialization.

User or Operator Response: None

System Programmer Response: Try to correct the problem with the help of the information in this message.

Source Data Set: EZASNACF.C

Procedure Name: add_trap_destination

EZZ6221I Need community name

Explanation: When the agent was started a -c argument was specified but there was no community name specified.

System Action: The agent terminated.

User or Operator Response: None.

System Programmer Response: A community name must be specified if the -c argument is specified. The community name follows the -c argument.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6222I Need a UNIX socket name

Explanation: When the agent was started a -s argument was specified but there was no UNIX socket name specified.

System Action: The agent terminated.

User or Operator Response: None.

System Programmer Response: A UNIX socket name must be specified if the -s argument is specified. The UNIX socket name follows the -s argument.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6223I Need timeout value

Explanation: When the agent was started a -t argument was specified but there was no time out value specified.

System Action: The agent terminated.

User or Operator Response: None.

System Programmer Response: A time value must be specified if the -t argument is specified. The time value is in seconds.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6224I *value* **is an incorrect timeout value, using** *timeout value*

Explanation: When the agent was started a -t argument was specified but the time value was not a valid number. The default time out value was used.

System Action: The agent continues initializing.

User or Operator Response: The user can terminate the agent and restart it with the intended time out value.

System Programmer Response: The user can terminate the agent and restart it with the intended time out value.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6225I SNMP agent: Initialization complete

Explanation: The SNMP agent has completed initialization and is ready to receive requests.

System Action: The agent is functioning.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6226I *port number* **is an incorrect port number, using** *default port*

Explanation: When the agent was started, a -p argument specified a *port number* which was not valid port. The default port value, *default port* is used instead.

System Action: The agent continues initializing.

User or Operator Response: The user can terminate the agent and restart it with the intended port number.

System Programmer Response: None.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6227I Added managers for community: *community name* **rc=** *rc*

Explanation: When the agent attempted to add the community name, it found that the name already existed. As a result, it becomes a multiple manager community name.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6228I (*error text***) from snmp_local_config_set()**

Explanation: The agent was attempting to set the DPI MIB variable, dpiPortForTCP, but was not successful. The *error text* will provide more information about the cause of the error. All subagents that use this connection will be unable to communicate with the agent. This does not affect unix stream subagent connections.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: set_dpiport

EZZ6229I Closing DPI UNIX socket connection, fd=*socket file descriptor*

Explanation: The agent closed a DPI UNIX socket connection to a sub-agent on the file descriptor *socket file descriptor*. This is normally a message that reports agent activity.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: close_dpi

EZZ6230I Closing DPI inet socket connection, fd=*socket file descriptor***, address** *ip address* **port** *port number*

Explanation: The agent closed a DPI inet socket connection to a sub-agent on the file descriptor *socket file descriptor ip address* is the ip address and the port number is *port number*. This is normally a message that reports agent activity.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: close_dpi

EZZ6231I (*error text*) from snmp_delete_subagent()

Explanation: The agent closed a DPI socket connection and then tried to to remove all control blocks related to this sub-agent but had a problem. The *rc* was returned from the snmp_delete_subagent function. The particular problem is specified by the additional *error text* error text.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: close_dpi

EZZ6232I The SNMP agent is running as jobname, *jobname*.

Explanation: The agent is reporting its jobname.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6233I Cannot allocate buffers for SNMP and DPI packets; SNMP agemt terminated.

Explanation: The agent cannot allocate enough space for buffers.

System Action: The agent terminates.

User or Operator Response: None.

System Programmer Response: Determine why there is not enough main memory.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6234I gethostname function failed; SNMP agent defaulting to 127.0.0.1

Explanation: The gethostname function failed. The agent will default to using the loopback address as its local address.

System Action: The agent continues

User or Operator Response: None.

System Programmer Response: Determine why the system cannot get its hostname if use of the loopback address isn't acceptable.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6235I *function name and socket type failure*; *error text*

Explanation: The failing *function name and socket type* The *error text* will provide more information about the cause of the error.

System Action: The agent may terminate if it involves SNMP socket(161).

User or Operator Response: None.

System Programmer Response: Determine why the system function failed.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6236I bind function failed for SNMP inet udp socket; *error text*

Explanation: The bind function failed to get a socket that would be used for communication between the agent and manager functions. The *error text* will provide more information about the cause of the error.

System Action: The agent terminates.

User or Operator Response: None.

System Programmer Response: Determine why the system cannot do a bind.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6237I *failing_socket_function_name for a socket_type*; *error text*

Explanation: The socket function failed for a socket that would be used for communication between the agent and sub-agents functions. The *error text* will provide more information about the cause of the error.

System Action: The agent may terminate depending on the failing function.

User or Operator Response: None.

System Programmer Response: Determine why the function failed.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6238I *socket function failed for UNIX DPI socket*; *error text*

Explanation: The *socket function failed for UNIX DPI socket* which is used for communication between the agent and sub-agents functions. The *error text* will provide more information about the cause of the error.

System Action: The agent may terminate depending on the socket function.

User or Operator Response: None.

System Programmer Response: Determine why the socket function is failing.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6239I selectex() function timed out on wait forever

Explanation: The selectex function completed on a time out; however, the selectex was called with a wait forever. This completion should never occur.

System Action: The agent continues if the count of these events is within a set limit.

User or Operator Response: This condition should be reported to your system administrator.

System Programmer Response: Contact the IBM Support Center for assistance.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6240I selectex() function failed return code = *rc*

Explanation: The selectex function failed with the return code shown.

System Action: The agent continues if the count of these events is within a set limit.

User or Operator Response: This condition should be reported to your system administrator.

System Programmer Response: Contact the IBM Support Center for assistance.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6241I receive function failed; *error text*

Explanation: The receive function failed. The *error text* will provide a more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6242I accept function failed for a DPI inet socket; *error text*

Explanation: The accept function failed for a DPI inet socket that would be used for communication between the agent and sub-agent functions. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6243I Refused new DPI connection, at maximum connection of *maximum_agents*

Explanation: The agent refused the connection of a new sub-agent because of a limit on the number of active sub-agents.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6244I Accepted new DPI inet socket connection on fd=*file descriptor* **from** *inet address* **port** *port number*

Explanation: The agent has accepted a connection to a sub-agent. This event is logged.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6245I accept function failed for a DPI UNIX socket; *error text*

Explanation: The accept function failed for a DPI UNIX socket that would be used for communication between the agent and sub-agent functions. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6246I Accepted new DPI UNIX socket connection on fd=*file descriptor***.**

Explanation: The agent has accepted a connection to a sub-agent over a UNIX socket connection. This event is logged.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6247I SNMP agent terminating, got *number* **errors on selectex**

Explanation: The selectex function failure count reached its limit.

System Action: The agent Terminates.

User or Operator Response: None.

System Programmer Response: Determine why the system is getting selectex failures.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6248I A select function timed out on a DPI connection

Explanation: The receive function failed. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: mustread

EZZ6249I EOF received on DPI fd *number*

Explanation: The receive function returned an EOF for the indicated file descriptor *number* while trying to receive from a subagent.

System Action: The DPI connection is closed. The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: read_tcp_dpi_packet

EZZ6250I receive function failed for a DPI socket; *error text*

Explanation: The receive function failed for a DPI socket. The *error text* indicates the specific reason for failure.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: read_tcp_dpi_packet

EZZ6251I sendto function failed for a DPI socket; *error text***.**

Explanation: The sendto function failed for a DPI socket. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: send_snmp_message

EZZ6252I On a sendto function only *number sent* **out of** *total number* **bytes sent.**

Explanation: The sendto function failed to send all the intended bytes on a DPI socket.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: asend_snmp_message

EZZ6253I send function failed; *error text*

Explanation: The send function failed on a DPI socket. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: send_message_to_snmp_subagent

EZZ6254I On a send function only *number* **out of** *number* **bytes sent**

Explanation: The send function failed to send all the intended bytes on a DPI socket.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: send_message_to_snmp_subagent

EZZ6255I select function failed return code = *rc*

Explanation: The select function failed. The return code is *rc*.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: await_message_from_snmp_subagent

EZZ6256I gethostbyname function failed for *hostname***;** *error text*

Explanation: The gethostbyname function failed. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: Determine if *hostname* is a valid hostname.

Source Data Set: EZASNAAG.C

Procedure Name: snmp_IPaddress LINKAGE lookup_host

EZZ6257I SIGPIPE received for an SNMP agent which is now shutting down

Explanation: A SIGPIPE signal was received by the agent.

System Action: The agent terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: sigpipedd

EZZ6258I SIGABND signal received. SNMP agent daemon terminating with *abendcode***,** *rsncode***.**

Explanation: A SIGABND signal was received by the agent.

System Action: The agent terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: pgmabndd

EZZ6259I Tracing is set to *trace setting*

Explanation: This a response to a modify command that indicates the current trace setting.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: mvs_command_handler

EZZ6260I Unrecognized modify request

Explanation: This a response to a modify command that could not be serviced because the request was not recognized.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: mvs_command_handler

EZZ6261I Modify request completed

Explanation: This a response to a modify command that completed.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: mvs_command_handler

EZZ6262I Unsupported command received

Explanation: The agent modify command handler does not support the requested modify.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAAG.C

Procedure Name: mvs_command_handler

EZZ6263I Configuration problem in file *filename* **at line** *line number* **variable name** *MIB variable name* **not recognized.**

Explanation: The agent was attempting to configure the MIB variables from the configuration file and did not recognize the MIB variable at the indicated line number.

System Action: The agent continues initializing.

User or Operator Response: None. at that line number.

System Programmer Response: Check the file and correct the MIB variable at that line number.

Source Data Set: EZASNAAG.C

Procedure Name: snmp_set_mib_install_defaults

EZZ6264I Configuration problem in file *file name* **at line** *line number* **variable name** *variable name* **the value:** *value* **not recognized.**

Explanation: The agent was attempting to configure the variable from the configuration file and did not recognize the variable at the indicated line number.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: Check the file and correct the MIB value at that line number.

Source Data Set: EZASNAAG.C

Procedure Name: snmp_set_mib_install_defaults

EZZ6265I Usage: *command* **[options], incorrect option specified. To see a complete list of options, enter** *command* **?**

Explanation: This is a list of all supported options.

In the message text:

options

-c *community* - a community name or passwrod for SNMP requests (default public).
-d *level* - debug level [0...255] default 0 (but 31 is used if -d is specified without a value)
-p *port* - listen for SNMP packets on this port (default 16)
-? display the usage statement.

System Action: Processing ends.

User or Operator Response: Correct the option or version and reissue the command.

System Programmer Response: None.

Source Data Set:

Procedure Name:

EZZ6266I Using file *file name* **for community names configuration**

Explanation: The agent is using the "file name" as its source of community name configuration.

System Action: None.

User or Operator Response: None.

System Programmer Response: The agent configures the community name from entries in this file.

Source Data Set: EZASNACF.C

Procedure Name: configure_v1_default

EZZ6267I Tracing set to *level*

Explanation: This message indicates what tracing level was requested. There may be several following lines in the syslog that further state each type of tracing requested.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6272I OE SNMP agent: Could not establish affinity with '*jobname***' (***errno***/***errno2* **)**

Explanation: The OE SNMP agent cannot communicate with the TCPIP stack *jobname*. The OE SNMP agent attempted to use the OE socket call, setibmopt(), to associate itself with the TCP/IP instance *tcpip_name*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance to which the OE SNMP agent is to be associated with. The setibmopt call failed with the displayed *error_code* and *reason*.

System Action: The OE SNMP agent ends abnormally.

User or Operator Response: Most likely, the TCP/IP instance's name was not defined correctly to OMVS. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure OMVS. Insure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used)

matches the SUBFILESYSTYPE NAME. Recycle OMVS or TCP/IP if a change is necessary. If none of the above error conditions exist contact your system programmer.

System Programmer Response: For the OE SNMP agent to communicate with a particular stack, the jobname (as determined by the system variable TCPIPjobname) must match "xxxxx" where "xxxxx" is set in the BPXPRMPx member used to start OMVS. "xxxxx" is set in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(EZBPFINI). In order to establish an affinity with a corresponding TCPIP stack, the OE SNMP agent uses the OE setibmopt call. Correct the error indicated by *error_code* and *reason*. For more information about errnos, see *OS/390 OpenEdition Programming Tools*.

Source Data Set: EZASNAAG.C

Procedure Name: main

EZZ6273I Value specified for the *startup* **parameter is missing**

Explanation: When the agent was started a - argument was specified without an value. The default value continues to be in effect.

System Action: The agent continues initializing.

User or Operator Response: The user can terminate the agent and restart it with the intended value.

System Programmer Response: The user can terminate the agent and restart it with the intended value.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

EZZ6274I *value* **is an invalid max retry value, using** *max retry value*

Explanation: When the agent was started a -r argument was specified but the max retry value was not a valid number. The default max retry value was used.

System Action: The agent continues initializing.

User or Operator Response: The user can terminate the agent and restart it with the intended max retry value.

System Programmer Response: The user can terminate the agent and restart it with the intended max retry value.

Source Data Set: EZASNACF.C

Procedure Name: snmp_config_parse_options

OTRACERT Messages

EZZ6354—EZZ6376

This section contains OTRACERT messages.

EZZ6354I OE Traceroute: *socket operation* **error detected (***errnolerrno2***)**

Explanation: OE Traceroute was unable to perform the indicated socket operation.

System Action: OE Traceroute halts and exits.

User or Operator Response: Collect debug information via the -d parameter and forward the results to your system programmer or administrator for resolution. Debug information can be saved by redirecting it to a file via the '>' operator.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: openSock, sendProbe, processTracerte

EZZ6357I OE Traceroute: Parameter *parameter* **not an integer**

Explanation: You specified a non-numeric parameter when a numeric parameter was expected.

System Action: The OE Traceroute will apply default and continue processing.

User or Operator Response: Cancel execution and correct parameter if default is not acceptable see, *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDNET

Procedure Name: optArgCh

EZZ6358I OE Traceroute: The value of *option* **must be between** *minvalue* **and** *maxvalue***.**

Explanation: You specified an incorrect *option* option value.

System Action: OE Traceroute uses default instead and continues processing.

User or Operator Response: Cancel execution if accepting default is not desired.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: optArgCh

EZZ6359I OE Traceroute: Incorrect -s (source address) specified - *option*

Explanation: You specified an incorrect source option.

System Action: OE Traceroute will ignore this parameter.

User or Operator Response: Stop execution if ignoring parameter is not desired.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6360I Usage: otracert <-q numProbes> <-p port> <-t tos> <-s srcAddr> <-n> <-v> <-w waitTime> <-m maxTTL> <-l> <-d> host <packetSize>

Explanation: Shows allowed syntax for calling OE Traceroute.

System Action: OE Traceroute exits after displaying usage information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6361I OE Traceroute: Extraneous parameter '*parameter***'.**

Explanation: You specified an extraneous parameter.

System Action: OE Traceroute halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter and resubmit the otracert command. For information about the otracert command, see *OS/390 TCP/IP OpenEdition: User's Guide*.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6362I OE Traceroute: Unknown parameter '*option***'.**

Explanation: You specified an unknown parameter

System Action: OE Traceroute halts and exits.

User or Operator Response: Check the unknown parameter for misspellings or other problems. Correct the option and resubmit the otracert command. For information about the otracert command, see *OS/390 TCP/IP OpenEdition: User's Guide*. Also consult related MAN pages and/or online help, if available.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6363I OE Traceroute: Host name too long '*name***'.**

Explanation: The field interpreted as the host name is too long. Host name must be less than or equal to 64 characters in length.

System Action: OE Traceroute halts and exits.

User or Operator Response: Correct the host name and resubmit the otracert command. For information about the otracert command, see *OS/390 TCP/IP OpenEdition: User's Guide*. Also consult related MAN pages and/or online help, if available.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6364I OE Traceroute: Allocation of probe failed - *packetSize***.**

Explanation: This error should not occur and indicates that Traceroute was unable to allocate the probe buffer.

System Action: OE Traceroute halts and exits.

User or Operator Response: Retry OE Traceroute with a smaller packetSize selection. For information about the otracert command, see *OS/390 TCP/IP OpenEdition: User's Guide*. Also consult related MAN pages and/or online help, if available.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6365I OE Traceroute: Host name or address not entered.

Explanation: A requested host name or IP address was missing.

System Action: The OE Traceroute halts and exits.

User or Operator Response: Reissue the otracert command with the host identification included.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: parsTRTE

EZZ6370I OE Traceroute: Failure detected in closing socket *socket* **(***errno***/***errno2***).**

Explanation: close() error

System Action: OE Traceroute exits.

User or Operator Response: Collect debug information via the -d parameter and forward the results to your system programmer or administrator for resolution. Debug information can be saved by redirecting it to a file via the '>' operator.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: processTracerte

OE Traceroute: unable to open message catalog 'otracertmsg.cat' - *strerror* **(***errno***/***errno2***)** **.**

Explanation: OE Traceroute was unable to open the traceroute message catalog "otracertmsg.cat" in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

System Action: OE Traceroute will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the IBM C/C++ for MVS/ESA Library Reference Volume 1 for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the OpenEdition MVS Advanced Application Programming Tools publication. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACDTRT

Procedure Name: main

EZZ6372I OE Traceroute: sigaction() failed for *signal* **:** *reason*

Explanation: OE Traceroute encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function	Description
SIGABND	Handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information on the abend will be lost and certain resources may not be properly cleaned up.
SIGTERM	Handler controls cleanup of resources during termination.
SIGPIPE	Handler allows the subagent to detect when the connection to OE Traceroute has been terminated.

System Action: Processing continues; however, the functions controlled by the failing signal will not function properly.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: main

EZZ6373I OE Traceroute: Unknown host *name***.**

Explanation: The host name specified could not be resolved to an IP address.

System Action: OE Traceroute halts and exits. Verify that host name resolution has been configured correctly.

User or Operator Response:

1. Correct the syntax of the host name and resubmit the otracert command. For information about the otracert command, see *OS/390 TCP/IP OpenEdition: User's Guide*.
2. Check that the specified host name is valid. If the host name looks correct, check with your system programmer to verify the host address.
3. Use the IP address, if it's known.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: main

EZZ6376I otracert was canceled

Explanation:

System Action: OE Traceroute was stopped by either user or operator action.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: termhand

Appendices

Appendix A. FTPD Reply Codes

This section contains FTPD Reply codes.

125 Appending to data set *name*

Severity: Preliminary.

Explanation: The current data set is being appended to the indicated data set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

125 command terminated due to server shutdown in progress

Severity: Preliminary.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

125 Could not allocate receive buffer

Severity: Preliminary.

Explanation: A command was issued to retrieve output from JES. A buffer required for this command to process is not available.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be sent.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

125 Data connection already open; transfer starting.

Severity: Preliminary.

Explanation: The FTP server is about to transfer data. The server determined that a data connection already existed and it was not necessary to open a new data connection. The FTP server will use the existing data connection for FTP transfer.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: data_connect

---

125 Data set *dsname* **is migrated and NoAutoRecall is specified.**

Severity: Preliminary.

Explanation: A submit job request was received for a job located in a data set at the server. The data set was migrated and needed to be recalled. The FTP server is currently in NoAutoRecall mode.

System Action: The command is rejected.

User or Operator Response: Issue the "SITE AUTORECALL" command to allow the data set to be recalled, and then re-issue the FTP command.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 *dsname* **is a physical sequential data set and a member was specified on the RETR command.**

Severity: Preliminary.

Explanation: The client entered the RETR command to request that a JES job be submitted and the output of the job retrieved. The pathname on the RETR command indicated that the job to be submitted was a member of the MVS partitioned data set *dsname*, but the FTP server determined that *dsname* was a physical sequential data set, and not a partitioned data set.

System Action: The command is rejected. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 Error allocating tape data set *dsname*

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a data set on a tape volume. The dynamic allocation was unsuccessful.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *alloc_tape*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

125 Error mounting volume

Severity: Preliminary.

Explanation: A request was received that requires a data set at the server. The data set will be allocated on a volume that is not mounted. The FTP server attempted to mount the volume but was unable to successfully do so.

System Action: The command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the volume cannot be mounted and correct the problem.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

125 Error mounting volume containing data set *dsname*

Severity: Preliminary.

Explanation: The volume containing the requested data set was not mounted to the FTP server system. The FTP server attempted to have the volume mounted, but was unable to get the volume mounted.

System Action: The command is rejected.

User or Operator Response: Re-issue the command. If the problem persists, contact the system programmer for the FTP server system.

System Programmer Response: Determine why the requested volume cannot be mounted to the MVS system.

Source Data Set: EZAFTPRD, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: list2, delete, renamefr, file_check, find_file, find_unique_file

---

125 Error mounting volume containing data set *dsname*

Severity: Preliminary.

Explanation: A request was received that requires a data set at the server. The data set is on a volume that is not mounted. The FTP server attempted to mount the volume but was unable to successfully do so.

System Action: The command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the volume cannot be mounted and correct the problem.

Source Data Set: EZAFTPMJ, EZAFTSRS

Procedure Name: jesFileCheck, find_file

---

125 Error recalling data set *dsname***.**

Severity: Preliminary.

Explanation: The command being processed requires a data set at the server. The data set was migrated and needed to be recalled. The FTP server attempted to recall the data set, but was unable to successfully recall it.

System Action: The command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the data set cannot be recalled and correct the problem.

Source Data Set: EZAFTPMJ, EZAFTPRS

Procedure Name: jesFileCheck, find_file

---

125 Error recalling data set *data_set* **(rc=***rc***)**

Severity: Preliminary.

Explanation: A RETR request was received for a data set. The data set was migrated and needed to be recalled. The FTP server attempted to recall the data set, but was unable to successfully recall the data set.

System Action: The RETR command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the data set cannot be recalled and correct the problem.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

125 File *dsname* **not found.**

Severity: Preliminary.

Explanation: The job was not submitted because the file was not found.

System Action: The job is not submitted. FTP continues.

User or Operator Response: Re-issue the request with a name of a file that can be found by the server. To ensure that the server can find the file, issue a DIR subcommand for the file after issuing SITE FILETYPE=SEQ.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

125 *file_name* **request nonexistent member to be sent.**

Severity: Preliminary.

Explanation: A command was issued against a member of a PDS. The member was not located in the PDS.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 Invalid data set name *"dsname"***. Use MVS Dsname conventions.**

Severity: Preliminary.

Explanation: The data set name violates one of the MVS file naming conventions and cannot be used to reference a data set at the server.

System Action: The job is not submitted. FTP continues.

User or Operator Response: Rename the data set in compliance with MVS data set naming conventions. For more information about MVS data set naming conventions, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

125 JES internal reader allocation failed

Severity: Preliminary.

Explanation: In preparation of submitting a job to MVS the Internal Reader needed to be allocated. The allocation of the Internal Reader failed.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

125 JES internal reader open failed

Severity: Preliminary.

Explanation: In preparation of submitting a job to MVS the Internal Reader needed to be opened. The request to open the Internal Reader failed.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

125 List started OK.

Severity: Preliminary.

Explanation: The LIST or NLST command is being executed. The server is getting ready to transfer the LIST or NLST command output back to the client.

System Action: The LIST or NLST command is executed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRD

Procedure Name: jesStatus, list2

125 List terminated.

Severity: Preliminary.

Explanation: The FTP server was preparing to transfer the output from the LIST or NLST command, but encountered an error and terminated the data transfer. This reply is followed by a 550 reply with additional information about the error.

System Action: The data transfer is terminated.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

125 Mismatched quotes on pathname *pathname*

Severity: Preliminary.

Explanation: The pathname provided for a data set that contains a job to submit was enclosed in an unbalanced set of single quotes.

System Action: The job is not submitted. FTP continues.

User or Operator Response: Enclose the data set name in balanced quotes -- a quote at the beginning and the end of the name.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

125 Name length error for pathname *pathname*

Severity: Preliminary.

Explanation: The pathname provided for a data set that contains a job to submit is too long. The name, when combined with the current working directory, must adhere to the following maximum lengths:

44 for a physical sequential data set
55 for a member of a PDS (includes the parentheses for the member name
1023 for a file in the hierarchical file system.

Note: The maximum length for a filename is 255 -- the total pathname maximum length is 1023.

System Action: The job is not submitted. FTP continues.

User or Operator Response: Re-issue the request with a name that meets the limits for the type of data set or file.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

125 Nlst started OK

Severity: Preliminary.

Explanation: A command was issued to obtain an NLST. The NLST command started ok.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be sent.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesStatus

125 Non-DASD data set *dsname* **cannot be processed.**

Severity: Preliminary.

Explanation: A command was issued against a data set that does not reside on a Direct Access Storage Device (DASD).

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

125 Open of *dsname* **failed.**

Severity: Preliminary.

Explanation: The data set or file containing a job cannot be opened.

System Action: The job is not submitted. FTP continues.

User or Operator Response: None.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The open file error will create a trace record with an errno code. Determine the cause of the error and correct the problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

125 Retrieve of a whole Partitioned data set is not supported. Use MGET for this purpose.

Severity: Preliminary.

Explanation: The client entered the RETR command to request that a JES job be submitted and the output of the job retrieved. The pathname on the RETR command indicated that the job to be submitted was the MVS data set *dsname*, but the FTP server determined that *dsname* was a partitioned data set, which requires a member name to be specified with *dsname*.

System Action: The command is rejected. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 Sending all SPOOL files for requested Jobid.

Severity: Preliminary.

Explanation: The FTP server is sending all the files in the spool for a specified job ID. For more information about spool files, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Action: The files are sent. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGEt

---

125 Sending data set *data set* **FIXrecfm** *record length*

Severity: Preliminary.

Explanation: The FTP server is sending the indicated data set with a fixed record length. The record length is indicated in the reply.

System Action: The data set is transferred. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

125 Sending data set *name*

Severity: Preliminary.

Explanation: The FTP server is sending the indicated data set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR

Procedure Name: jesGet, retrieve

---

125 Sending file via NJE to requested destination.

Severity: Preliminary.

Explanation: The data that is transferred from the client is being sent via Network Job Entry (NJE) to the destination specified by the SITE DEST= parameter.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

125 Sending Job to JES internal reader FIXrecfm *record length*

Severity: Preliminary.

Explanation: The FTP server is sending the indicated job to the job entry system (JES) internal reader with a fixed record length. The record length is indicated in the reply. For more information on JES, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Action: The job is sent to JES. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

---

125 Sending Job to JES internal reader VARrecfm *record length*

Severity: Preliminary.

Explanation: The FTP server is sending the indicated job to the job entry system (JES) internal reader with a variable record length. The record length is indicated in the reply. For more information on JES, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Action: The job is sent to JES. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

---

125 Storing data in the Null directory (*dev.null).

Severity: Preliminary.

Explanation: The FTP server is transferring data from the client but is not storing the data in the file system. This action was requested with a change directory subcommand to the null directory.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_null_dir

---

125 Storing data set *name*

Severity: Preliminary.

Explanation: The FTP server is storing the indicated data set. If the data set already exists, it is replaced by the transferred data.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

125 Storing data set *name* **(unique name)**

Severity: Preliminary.

Explanation: The FTP server is storing the indicated data set with a unique name. This prevents overwriting or erasure of existing files on the remote host.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

125 Submit fails: *dsname* **User not authorized.**

Severity: Preliminary.

Explanation: The job was not submitted. The requested data set is protected by a security system such as RACF, and the user is not authorized to read the data set.

System Action: The job is not submitted. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to read the data set.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

125-Submitting job *job_id* **FIXrecfm** *record_length*

Severity: Preliminary.

Explanation: The indicated job is being submitted to the Job Entry System (JES).

System Action: The job is submitted. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

125-Submitting job *job_id* **VARrecfm** *record_length*

Severity: Preliminary.

Explanation: The indicated job is being submitted to the Job Entry System (JES).

System Action: The job is submitted. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

125 Transfer aborted: file error.

Severity: Preliminary.

Explanation: A file error occurred while processing the command.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

125 Transfer aborted: send error.

Severity: Preliminary.

Explanation: While attempting to submit a job from the command, the data connection was lost. The job may or may not have been successfully submitted to the system for processing.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

125 Transfer request aborted

Severity: Preliminary.

Explanation: The FTP server closed the data connection because of an internal error. This reply terminates an outstanding preliminary reply.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: req_cleanup

---

125 Unable to get Jobid

Severity: Preliminary.

Explanation: GET command with automatic retrieve was issued. The jobid could not be obtained after being submitted.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be sent.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

125 Unable to send *name*

Severity: Preliminary.

Explanation: The file used in a JesPutGet operation either cannot be found, is being used by another process, or cannot be opened.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 User Exit refuses this Job to be submitted by *dsname*

Severity: Preliminary.

Explanation: During the submit of a job to JES, a JES user or installation exit encountered an error. The request to submit the job may have failed.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

125 Volume containing *dsname* **is not mounted and NoAutoMount specified."**

Severity: Preliminary.

Explanation: A submit job request was received for a job located in a data set at the server. The data set is on a volume that is not mounted. The FTP server is currently in NoAutoMount mode.

System Action: The command is rejected.

User or Operator Response: Issue the "SITE AUTOMOUNT" command to allow the volume to be mounted, and then re-issue the FTP command.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

---

125 Volume is not ready and automatic tape mounts are not allowed.

Severity: Permanent negative completion.

Explanation: A request to retrieve a data set from or store a data set on a tape volume was received. The tape volume is not mounted and the end user has requested NoAutoTapeMount.

System Action: The request is not performed.

User or Operator Response: Request that the tape be mounted by the operator and then re-issue the request. Also, the SITE AUTOTAPEMOUNT subcommand can be issued to allow automatic tape mounts to occur.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

125 VSAM data set *dsname* **cannot be processed.**

Severity: Preliminary.

Explanation: A command was issued that requests action be performed against a VSAM data set. VSAM data sets are not supported.

System Action: FTP server continues normal execution. The FTP client waits for the next reply to be presented by the server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck

125-Waiting for allocation of tape data set *dsname*

Severity: Preliminary.

Explanation: A retrieve or store command was issued for a data set that is on a tape volume. The FTP server may have to have the volume mounted. The server will wait for the volume to be successfully mounted and then begin processing the data set.

System Action: The FTP server prompts the operator to mount the volume. FTP processing continues once the volume is mounted.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

125-Waiting for recall of data set *data set*

Severity: Preliminary.

Explanation: A command was issued for a data set that is migrated. The FTP server must recall the data set in order complete the command. The server waits for the data set to be successfully recalled and then begins processing the command.

System Action: The FTP server recalls the data set. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRD, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: jesFileCheck, list2, renamefr, file_check, find_file, find_unique_file

---

125-Waiting for volume mount for *dsname*

Severity: Preliminary.

Explanation: A command was issued for a data set that is catalogued on a volume which is not mounted. The FTP server must mount the volume in order to complete the command. The server waits for the volume to be successfully mounted and then begins processing the command.

System Action: The FTP server prompts the operator to mount the volume. FTP processing continues once the volume is mounted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRD, EZAFTPRR

Procedure Name: jesFileCheck, list2, file_check

125-Waiting for volume mount for volume *volume*

Severity: Preliminary.

Explanation: The FTP server is waiting for the volume to be mounted. The FTP reply code of 125 in the reply indicates that the file status is okay and the FTP server is about to open the data connection.

System Action: FTP continues. The FTP server mounts the volume automatically.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMV, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: seq_create_file, renamefr, file_check, find_file, find_unique_file

---

125 When *job_id* **is done, will retrieve its output.**

Severity: Preliminary.

Explanation: When the indicated job is completed, the FTP server will retrieve the output from the Job Entry System (JES).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

150 command terminated due to server shutdown in progress

Severity: Preliminary.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

150 Waiting for allocation of tape data set *dsname*

Severity: Preliminary.

Explanation: A retrieve or store command was issued for a data set that is on a tape volume. The FTP server may have to have the volume mounted. The server will wait for the volume to be successfully mounted and then begin processing the data set.

System Action: The FTP server prompts the operator to mount the volume. FTP processing continues once the volume is mounted.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

150 Waiting for recall of data set *dsname*

Severity: Preliminary.

Explanation: The FTP server is waiting for the recall of the indicated data set. The FTP reply code of 150 in this reply indicates that the file status is okay and the FTP server will open a data connection for the data transfer.

System Action: The FTP server recalls the data set. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: renamefr, file_check, find_file, find_unique_file

---

150 Waiting for volume mount for *dsname*

Severity: Preliminary.

Explanation: The FTP server is waiting for the volume containing the indicated data set to be mounted. The FTP reply code of 150 in the reply indicates that the file status is okay and the FTP server is about to open the data connection.

System Action: FTP continues. The FTP server mounts the volume automatically.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMV, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: seq_create_file, renamefr, file_check, find_file, find_unique_file

---

200- % Free Free Largest Free
200- Volume Unit Free Cyls Trks Cyls-Trks Exts Address Use Attr

Severity: Completion.

Explanation: These replies display the headers for the information provided when the QDISK parameter has been specified for the SITE subcommand.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: qdiskp

200-Alternate languages are not available

Severity: Completion.

Explanation: A SITE LANG command was issued to change the language being used for FTP replies, but no reply languages were established in the FTP.DATA file, so the current language is the only language available.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: If other reply languages are desired, use the REPLYLANGUAGE statement in the FTP.DATA file to define the languages and their respective catalogs to the FTP server. Refer to the *OS/390 TCP/IP OpenEdition User's Guide* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BLOCKSIZE being set to *blksize*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values caused a mismatch between the LRECL, RECFM, and BLKSIZE parameters. The BLKSIZE value is reset to a compatible value, *blksize*.

System Action: The BLKSIZE parameter is reset to *blksize*.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL and RECFM combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BLOCKSIZE must be a multiple of LRECL for RECFM *recfm*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values caused a mismatch between the LRECL, RECFM, and BLKSIZE parameters. The Record Format, *recfm*, requires that BLKSIZE be a multiple of LRECL, but the BLKSIZE was not a multiple of LRECL.

System Action: FTP attempts to set the parameters to compatible values. This reply is followed by another 200- reply indicating which values have been reset.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL, RECFM, and BLKSIZE combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BLOCKSIZE must be at least 4 more than LRECL for RECFM *recfm*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values caused a mismatch between the LRECL, RECFM, and BLKSIZE parameters. The Record Format, *recfm*, requires that BLKSIZE be at least 4 more than LRECL, but the BLKSIZE was not at least 4 more than LRECL.

System Action: FTP attempts to set the parameters to compatible values. This reply is followed by another 200- reply indicating which values have been reset.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL, RECFM, and BLKSIZE combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BLOCKSIZE must equal LRECL for RECFM *recfm*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values caused a mismatch between the LRECL, RECFM, and BLKSIZE parameters. The Record Format, *recfm*, requires that LRECL and BLKSIZE be equal, but the values for BLKSIZE and LRECL were not the same.

System Action: FTP attempts to set the parameters to compatible values. This reply is followed by another 200- reply indicating which values have been reset.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL, RECFM, and BLKSIZE combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Blocksize parameter *(value)* **is not numeric. Blocksize ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the BLKSIZE parameter, but the value, *value*, specified for the BLKSIZE parameter was not a numeric value. The value for BLKSIZE must be a numeric value between 1 and 32760.

System Action: The BLKSIZE parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Blocksize parameter *(value)* **must be between 1 and 32760. Blocksize ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the BLKSIZE parameter, but the value, *value*, specified for the BLKSIZE parameter was outside the valid range. The value for the BLKSIZE parameter must be a numeric value between 1 and 32760.

System Action: The BLKSIZE parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BUfno parameter *(value)* **is not numeric. BUfno ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the BUFNO parameter, but the value, *value*, specified for the BUFNO parameter was not a numeric value.

System Action: The BUFNO parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-BUfno parameter *(value)* **must be between 1 and 255. BUfno ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the BUFNO parameter, but the value, *value*, specified for the BUFNO parameter was outside the valid range.

System Action: The BUFNO parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

202 ACCT command not required - ignored.

Severity: Completion.

Explanation: The FTP server received the ACCT command, which is not supported under MVS. The command is ignored.

System Action: FTP continues.

User or Operator Response: Use the USER command instead of the ACCT command. For more information about the USER command, see the *OS/390 TCP/IP OpenEdition User's Guide*, or type HELP USER at the command line.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: nosyntchk_cmd

200-CHKPTInt parameter *(value)* **is not numeric. Chkptint ignored.**

Severity: Completion.

Explanation: The value of the CHKPTInt parameter, *value*, was not a numeric value, or was outside the acceptable range.

System Action: The parameter is ignored.

User or Operator Response: Reissue the SITE command with a valid value specified for the CHKPTInt parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-CHKPTInt parameter *(value)* **is larger than a 4 byte integer. Chkptint ignored.**

Severity: Completion.

Explanation: The value of the CHKPTInt parameter, *value*, was larger than the maximum value allowed for a 4 byte integer.

System Action: The value in error is ignored.

User or Operator Response: Reissue the SITE command with a valid value specified for the CHKPTInt parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod cannot access file *filename* **:** *error*

Severity: Completion.

Explanation: The FTP server encountered an error attempting to obtain the file information for *filename*. *error* is the C runtime library error encountered while accessing the file.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Reissue the SITE command with a valid value specified for the filename on the CHMOD parameter. If the problem persists, contact the system programmer.

System Programmer Response: If necessary, correct the error indicated by *error*.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod failed: *error*

Severity: Completion.

Explanation: The FTP server encountered an error attempting to change the permission bits of the file. *error* is the C runtime library error returned in response to the chmod() request.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error indicated by *error*

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod filename is not a valid HFS file. CHMOD ignored

Severity: Completion.

Explanation: The filename specified on the CHMOD parameter of the SITE subcommand was not an HFS file. The CHMOD parameter is only valid for HFS files.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Reissue the SITE command with a valid value specified for the filename on the CHMOD parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod invalid filename *filename*

Severity: Completion.

Explanation: The *filename* specified by the CHMOD parameter of the SITE subcommand was longer than the HFS filename maximum of 1023 characters. If the filename operand of the CHMOD parameter began with a forward slash (/), then *filename* will be the filename operand used "asis"; otherwise *filename* will be the filename operand appended to the current working directory.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Reissue the SITE command with a valid value specified for the filename on the CHMOD parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod invalid syntax. Specify CHMOD mode filename. CHMOD ignored

Severity: Completion.

Explanation: The Chmod parameter of the SITE command was entered incorrectly. The correct syntax of the CHMOD parameter of the SITE subcommand is SITE CHMOD *mode filename* where *mode* is the new permission bit setting for the file, and *filename* is the name of the file to change.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Reissue the SITE command with the correct syntax for the CHMOD parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod mode value invalid. Specify mode as a 1-3 digit octal value, or as {u|g|o|a}{=|+|-}{r|w|x|rw|rx|wx|rwx}. CHMOD ignored

Severity: Completion.

Explanation: The SITE command was entered with the CHMOD parameter, but the mode operand of the CHMOD parameter was incorrect. The mode operand specifies the permission bit settings of the file, and should be expressed as either a 1-3 digit octal number (for example, `666` for permission setting rw-rw-rw-), or as a mnemonic indicating the changed bits, (for example `a+x` to turn on the execute bit for user, group, and other).

System Action: The CHMOD parameter is ignored.

User or Operator Response: Re-enter the SITE subcommand, specifying the correct value for the mode operand. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod not allowed when user not logged in. CHMOD ignored.

Severity: Completion.

Explanation: The SITE subcommand was issued with the CHMOD parameter, but the user was not currently logged on to the server with a valid userid. The user must first login with a valid userid and password before issuing the SITE subcommand with the CHMODE parameter.

System Action: FTP continues.

User or Operator Response: Login to the server using the USER and PASS subcommands, then reissue the SITE subcommand with the CHMOD parameter.

System Programmer Response:

Source Data Set: EZAFTPMS

Procedure Name: site

200-Chmod parameters missing. Specify CHMOD mode filename. CHMOD ignored

Severity: Completion.

Explanation: The SITE command was entered with the CHMOD parameter, but the syntax of the CHMOD parameter was incorrect. One or more of the operands required on the CHMOD parameter were missing. The syntax of the CHMOD parameter is SITE CHMOD *mode filename*.

System Action: The CHMOD parameter is ignored.

User or Operator Response: Re-enter the SITE subcommand, specifying the correct operands. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

200-Conddisp parameter *(value)* **is invalid. Conddisp ignored.**

Severity: Completion.

Explanation: The SITE command was entered with the CONDDISP parameter, but the value specified, *value*, was not a valid value. Valid values for the CONDDISP parameter are Catlg and Delete.

System Action: The CONDDISP parameter is ignored.

User or Operator Response: Re-enter the SITE subcommand, specifying the correct value for CONDDISP. Refer to the *OS/390*

TCP/IP OpenEdition User's Guide, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Ctrlconn invalid syntax. Specify CTRLCONN=ascii_code_page_name or CTRLCONN=7BIT.

Severity: Completion.

Explanation: A SITE subcommand has been entered with a CTRLCONN parameter whose value begins with '('. The syntax for the CTRLCONN parameter does not use parentheses. The CTRLCONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Reenter the SITE command with corrected syntax.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Data set " *dsname*" does not exist. Dcbdsn parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, could not be found.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Reissue the SITE DCBDSN subcommand, specifying a valid MVS data set.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-" *dsname*" invalid dsorg. Dcbdsn parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, did not have a valid data set organization, or the FTP server was not able to determine the data set organization of the data set. Only MVS PS and PDS data sets can be used as a model DCB data set by the FTP server.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Reissue the SITE DCBDSN subcommand, specifying a valid MVS data set.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-" *dsname*" is a VSAM data set. Dcbdsn parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was a Virtual Storage Access Method (VSAM) data set. Only MVS PS and PDS data sets can be used as a model DCB data set by the FTP server.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Reissue the SITE DCBDSN subcommand, specifying a valid MVS data set.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-" *dsname*" is migrated and noautorecall is specified. DCBDSN parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was migrated and the FTP server could not recall the dataset because NOAUTORECALL had been specified at the server.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Issue the SITE subcommand with the AUTORECALL parameter to allow the dataset to be recalled.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-" *dsname*" is not on a direct access volume. Dcbdsn parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was not located on a direct access volume. The model DCB data set must be located on a direct access volume to be used by the FTP server.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Reissue the SITE DCBDSN subcommand, specifying a valid MVS data set.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Data set name missing from DCBDSN parameter. DCBDSN parameter ignored.

Severity: Completion.

Explanation: The DCBDSN parameter of the SITE subcommand was specified with an equal sign, but no data set name followed the equal sign.

System Action: The DCBDSN parameter is ignored. FTP continues.

User or Operator Response: If the SITE DCBDSN= was issued to setup a model DCB data set, reissue the command specifying a data set after the equal sign. If the SITE DCBDSN= was issued to clear the setting of the model DCB data set, reissue the command with no

equal sign. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Data transfer is mixed SBCS/DBCS

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. The current transfer type has now been changed to transmit mixed SBCS/DBCS data.

System Action: Program execution continues. The current FTP transfer type now uses mixed SBCS/DBCS data.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

200 Data transfer is pure DBCS

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. The current transfer type has now been changed to transmit only pure DBCS data. No SO/SI characters will be present in EBCDIC DBCS data.

System Action: Program execution continues. The current FTP transfer type now treats EBCDIC data as pure DBCS data with no SO/SI characters.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

200 Data transfer mode is Block.

Severity: Completion.

Explanation: The MODE command was entered to change the data transfer mode to Block. The request is processed successfully.

System Action: The data transfer mode is changed to Block. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: mode

---

200 Data transfer mode is Compressed.

Severity: Completion.

Explanation: The MODE command was entered to change the data transfer mode to Compressed. The request is processed successfully.

System Action: The data transfer mode is changed to Compressed. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: mode

---

200 Data transfer mode is Stream.

Severity: Completion.

Explanation: The MODE command was entered to change the data transfer mode to Stream. The request is processed successfully.

System Action: The data transfer mode is changed to Stream. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: mode

200-Dataclass parameter *(value)* **is more than 8 characters. Dataclass ignored.**

Severity: Completion.

Explanation: The value, *value*, specified for the Dataclass parameter is not valid. The Dataclass parameter value cannot be more than 8 characters long.

System Action: The Dataclass parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the DATACLASS parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-DB2 parameter *(db2name)* **is more than 4 characters. DB2 ignored.**

Severity: Completion.

Explanation: The *db2name* specified for the DB2 parameter on an FTP SITE command is too long.

System Action: The DB2 parameter is ignored. FTP continues.

User or Operator Response: Correct the name of the DB2 subsystem, and issued the SITE command again.

System Programmer Response: None

Source Data Set: EZAFTPMS

Procedure Name: site

200-Dest node longer than 8 characters. Dest ignored.

Severity: Completion.

Explanation: The DEST parameter of the site command was specified with a userid and node, and the node operand was longer than 8 characters. The node operand must be 8 characters or less.

System Action: The DEST parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the DEST parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Dest user longer than 8 characters. Dest ignored.

Severity: Completion.

Explanation: The DEST parameter of the site command was specified with a userid and node, and the userid operand was longer than 8 characters. The userid operand must be 8 characters or less.

System Action: The DEST parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the DEST parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Destination *(value)* is more than 8 characters. Dest ignored.

Severity: Completion.

Explanation: The destination, *value*, specified for the DEST parameter is not valid. The Dest parameter destination operand cannot be more than 8 characters long.

System Action: The Dest parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the DEST parameter.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Directory parameter *(value)* is not numeric. Directory ignored.

Severity: Completion.

Explanation: The SITE command was specified with the DIRECTORY parameter, but the value, *value*, specified for the DIRECTORY parameter was not a numeric value.

System Action: The DIRECTORY parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Directory parameter *(value)* must be between 1 and 16777215. Directory ignored.

Severity: Completion.

Explanation: The SITE command was specified with the DIRECTORY parameter, but the value, *value*, specified for the DIRECTORY parameter was outside the valid range.

System Action: The DIRECTORY parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Error locating file " *dsname*". DCBDSN parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The FTP server was unable to locate the data set, *dsname*, which was specified as the model DCB data set.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Reissue the SITE DCBDSN subcommand, specifying a valid MVS data set.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Error mounting " *dsname*". DCBDSN parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was on a volume which was not mounted to the system, and the FTP server encountered an error attempting to have the volume mounted.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Contact the system programmer

System Programmer Response: Determine why the volume cannot be mounted and correct the error.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Error retrieving " *dsname*". DCBDSN parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was migrated, and the FTP server encountered an error attempting to retrieve the dataset.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Contact the system programmer

System Programmer Response: Determine why the data set cannot be recalled and correct the error.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Filetype parameter *(value)* is invalid. Filetype ignored.

Severity: Completion.

Explanation: The value, *value*, specified for the FILETYPE parameter is not valid. FILETYPE must be SEQ, JES, or SQL.

System Action: The FILETYPE parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the FILETYPE parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Invalid format data set name " *name*". Dcbdsn parameter ignored.

Severity: Completion.

Explanation: The data set name specified for the DCBDSN parameter of the SITE command, which is used to specify the name of the data set to be used as a model for allocation of new data sets, has an incorrect format. The data set name must conform to MVS data set naming conventions.

System Action: The DCBDSN parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid MVS data set name. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS, EZAFTPRU

Procedure Name: site, check_dcbds

200-Invalid record format '*value*' encountered.

Severity: Completion.

Explanation: The SITE subcommand encountered an invalid internal value for the record format.

System Action: FTP attempts to reset the BLKSIZE, RECFM, and LRECL parameter values back to the default values.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Jeslrecl parameter *(value)* is not numeric. Jeslrecl ignored.

Severity: Completion.

Explanation: The SITE command was specified with the JESLRECL parameter, but the value, *value*, specified for the JESLRECL parameter was not a numeric value. The value for the JESLRECL parameter must be a numeric value between 1 and 254, or an '*'.

System Action: The JESLRECL parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-JesLrecl parameter must be between 1 and 254. JesLrecl parameter ignored.

Severity: Completion.

Explanation: The value submitted for the JESLRECL parameter, which is used to specify the logical record length for the JES internal reader at the remote host, was not within the valid range of 1 through 254.

System Action: The JESLRECL parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command specifying value for the JESLRECL parameter in the valid range of 1 to 254. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Jesrecfm parameter *(parameter)* is invalid. Jesrecfm ignored.

Severity: Completion.

Explanation: The value specified for the record format of the Job Entry Subsystem (JES) internal reader, JESRECFM, is not a valid value for that parameter. JESRECFM must be F, V, or *.

System Action: The JESRECFM value is ignored. FTP continues.

User or Operator Response: Reissue the SITE command specifying a valid value for JESRECFM. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as required.

Source Data Set: EZAFTPMS

Procedure Name: site

200- *keyword*=*value* **ignored. Unable to set up requested conversion.**

Severity: Completion.

Explanation: A CTRLCONN parameter was entered on a SITE subcommand, but FTP was unable to build the single-byte translate tables for the control connection. The CTRLCONN parameter is ignored. The translate tables for the control connection are not changed.

Note: If the FTP server is running in a double-byte environment, the CTRLCONN parameter cannot be used to change the translate tables for the control connection.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: If the FTP server is running in a double-byte code set, the CTRLCONN parameter is not supported. The tables used for the control connection are set at server initialization from either a TCPXLBIN file, or from FTP's internal default single-byte tables.

If the server is running in a single-byte environment, an internal error has occurred. If this occurs repeatedly, contact the IBM Support Center with an FTP trace.

Source Data Set: EZAFTPMS

Procedure Name: site

200- *language* **language is not available.**

Severity: Completion.

Explanation: A SITE LANG command was issued to change the language being used for FTP replies to *language*, but *language* is not one of the available languages.

System Action: FTP continues.

User or Operator Response: Use the SITE QLANG command to display a list of the available languages. The language requested by the SITE LANG command must be one of those listed by the SITE QLANG command. (The spelling must be exactly the same, although the case may differ.)

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Local byte size is 8

Severity: Completion.

Explanation: The FTP TYPE L command has been accepted, and the data transfer byte-size is 8 bits.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-LRECL being changed to *lrecl*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values caused a mismatch between the LRECL, RECFM, and BLKSIZE parameters. The LRECL value is changed to a compatible value, *lrecl*.

System Action: The LRECL parameter is reset to *lrecl*.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL, RECFM, and BLKSIZE combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-LRECL being reset to *lrecl*

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the BLKSIZE, LRECL, and RECFM parameters, and the resulting parameter values were incompatible The LRECL value is reset to a compatible value, *lrecl*.

System Action: The LRECL parameter is reset to the value it had prior to the SITE subcommand, if compatible, otherwise it is reset to the value from FTP.DATA, if compatible, otherwise it is set to the default.

User or Operator Response: Reissue the SITE subcommand with a valid LRECL, RECFM, and BLKSIZE combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-LRECL being reset to site default of 'not used'.

Severity: Completion.

Explanation: The SITE subcommand was entered with one or more of the LRECL, RECFM, and BLKSIZE parameters, and the resulting parameter values were incompatible. The previous LRECL value was also not compatible and the LRECL was reset to the value specified in the FTP.DATA file. The value in the FTP.DATA file was a NULL value, indicating that the LRECL should be unspecified.

System Action: The LRECL parameter is reset to the NULL value from the FTP.DATA file.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid BLKSIZE, LRECL, and RECFM combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Lrecl parameter *(value)* **is not numeric. Lrecl ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the LRECL parameter, but the value, *value*, specified for the LRECL parameter was not a numeric value. The value for the LRECL parameter must be a numeric value between 0 and 32760.

System Action: The LRECL parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Lrecl parameter *(value)* **must be between 0 and 32760. Lrecl ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the LRECL parameter, but the value, *value*, specified for the LRECL parameter was outside the valid range. The value for the LRECL parameter must be a numeric value between 0 and 32760.

System Action: The LRECL parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-LRECL X valid only for RECFM of U or variable length spanned.

Severity: Completion.

Explanation: The SITE subcommand was entered with the LRECL and/or RECFM parameters, and the resulting parameter values caused a mismatch between the LRECL and RECFM parameters. The LRECL value was X, but the RECFM was not U or variable length spanned. Logical record length of X is only valid when the record format is U, VS, VSA, VSM, VBS, VBSA, or VBSM.

System Action: FTP attempts to reset the values to compatible values. This reply is followed by another 200- reply indicating which values have been reset.

User or Operator Response: Reissue the SITE subcommand with a valid LRECL and RECFM combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-LRECL 0 valid only for RECFM of U.

Severity: Completion.

Explanation: The SITE subcommand was entered with the LRECL and/or RECFM parameters, and the resulting parameter values caused a mismatch between the LRECL and RECFM parameters. The LRECL value was 0, but the RECFM was not U. Logical record length of 0 is only valid when the record format is U.

System Action: FTP attempts to set the parameters to compatible values. This reply is followed by another 200- reply indicating which values have been reset.

User or Operator Response: If necessary, reissue the SITE subcommand with a valid LRECL and RECFM combination. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-LRECL, RECFM, and BLOCKSIZE being reset to default values.

Severity: Completion.

Explanation: The SITE subcommand encountered an unresolvable incompatibility between the RECFM, LRECL and BLKSIZE parameter values and has reset all three parameter values back to the default.

System Action: FTP attempts to reset the BLKSIZE, RECFM, and LRECL parameter values back to the default values.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Mgmtclass parameter *(value)* **is more than 8 characters. Mgmtclass ignored.**

Severity: Completion.

Explanation: The value, *value*, specified for the Mgmtclass parameter is not valid. The Mgmtclass parameter value cannot be more than 8 characters long.

System Action: The Mgmtclass parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the MGMTCLASS parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Migratevol parameter *(value)* **is more than 6 characters. Migratevol ignored.**

Severity: Completion.

Explanation: The value, *value*, specified for the Migratevol parameter is not valid. The Migratevol parameter value cannot be more than 6 characters long.

System Action: The Migratevol parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the MIGRATEVOL parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-NCP parameter not supported for the C server. NCP ignored.

Severity: Completion.

Explanation: The SITE command was entered with the NCP parameter, but the NCP parameter is not supported for the FTP C server.

System Action: The NCP parameter is ignored. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-No conversion available between *parm1* **and** *parm2***. SBDATACONN ignored.**

Severity: Completion.

Explanation: A SITE subcommand has been entered with an SBDATACONN parameter, but there is no supported code set converter for the code sets *parm1* and *parm2*. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Refer to the *OS/390 C/C++ Programming Guide* for information on supported code set converters and valid code set names. Resubmit the corrected SITE subcommand, with the EBCDIC code set name as the first SBDATACONN value, followed by the ASCII code set name.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-No other languages are available for replies.

Severity: Completion.

Explanation: A SITE QLANG command was issued to qyery the languages that are available for reply texts. Either no reply languages were defined to the server in the FTP.DATA file, or none of the defined reply catalogs are currently available to the server.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: If other reply languages are desired, use the REPLYLANGUAGE statement in the FTP.DATA file

to define the languages and their respective catalogs to the FTP server, and ensure that the translated catalogs are stored in the specified directories. Refer to the TCP/IP V3 for OpenEdition MVS Applications Feature Guide, FTP chapter, for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPMS

Procedure Name: site

200-No storage volumes exist.

Severity: Completion.

Explanation: The QDISK parameter was entered without a specific volume serial number. When this parameter is left blank, statistics about all storage volumes are displayed. No storage volumes were found.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: qdiskp

200-NORDW parameter not supported for the C server. NORDW ignored.

Severity: Completion.

Explanation: The SITE command was entered with the NORDW parameter, but the NORDW parameter is not supported for the FTP C server.

System Action: The NORDW parameter is ignored. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 OK

Severity: Completion.

Explanation: A NOOP command has been executed successfully, indicating that the remote host is still responding. For more information about the NOOP command, see the *OS/390 TCP/IP OpenEdition User's Guide* or type HELP NOOP on the command line.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: noarg_cmd

200-Operand not allowed on *parm* parameter. Parameter ignored.

Severity: Completion.

Explanation: The parameter *parm* appeared on a SITE subcommand in incorrect format. No operand is allowed for this parameter. The parameter is ignored.

System Action: FTP continues.

User or Operator Response: If desired, reissue the SITE command without an operand on the *parm* parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-operand required on *parm* parameter. Parameter ignored.

Severity: Completion.

Explanation: The parameter *parm* appeared on a SITE subcommand without an operand. The parameter is ignored.

System Action: FTP continues.

User or Operator Response: If desired, reissue the SITE command with an operand for the *parm* parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200- *parameter* **ignored.** *parameter* **already specified on this Site command.**

Severity: Completion.

Explanation: *parameter* was specified on the SITE command, but *parameter* was already encountered on this SITE command. A parameter can only be issued once per SITE command. All occurrences of the parameter after the first occurrence are ignored.

System Action: *parameter* is set to the value specified the first time the parameter was encountered.

User or Operator Response: If you wish to change the parameter to a specification that was not the first one on the SITE command, reissue the SITE command with the appropriate parameters.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200- *parameter*=*operand* **ignored. Requested conversion is not supported.**

Severity: Completion.

Explanation: A CTRLCONN paramter was entered on a SITE subcommand, but there is no conversion available between the FTP server's code page and *operand*. The CTRLCONN parameter is ignored. The translate tables for the control connection are not changed.

System Action: FTP continues.

User or Operator Response: Reissue the SITE command, specifying either "7bit" or the name of an ASCII code page supported by iconv. Refer to the *OS/390 C/C++ Programming Guide* for a list of the code pages supported by iconv. The code page name must be entered exactly as shown in the list (for example: IBM-850 or ISO8859-1).

System Programmer Response: None

Source Data Set: EZAFTPMS

Procedure Name: site

200- *parameter=value* **ignored.** *parameter* **already specified on this Site command.**

Severity: Completion.

Explanation: *parameter=value* was specified on the SITE command, but *parameter* was already encountered on this SITE command. A parameter can only be issued once per SITE command. All occurrences of the parameter after the first occurrence are ignored.

System Action: *parameter* is set to the value specified the first time the parameter was encountered.

User or Operator Response: If you wish to change the parameter value to the value specified by *parameter=value*, re-issue the site command with *parameter=value* as the only occurrence of the parameter on the SITE command.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200- *parameter1* **ignored.** *parameter2* **already specified on this Site command.**

Severity: Completion.

Explanation: The SITE command parameters *parameter1* and *parameter2* are mutually exclusive and cannot be specified on the same SITE command. *parameter2* has already been encountered on this SITE command, causing *parameter1* to be rejected.

System Action: *parameter1* is not changed. FTP continues.

User or Operator Response: If you wish to change *parameter1*, enter it on a SITE command without *parameter2*. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Port request OK.

Severity: Completion.

Explanation: The PORT command has been accepted.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: port

200-Primary parameter *(value)* **is not numeric. Primary ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the PRIMARY parameter, but the value, *value*, specified for the PRIMARY parameter was not a numeric value.

System Action: The PRIMARY parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Primary parameter *(value)* **must be between 1 and 16777215. Primary ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the PRIMARY parameter, but the value, *value*, specified for the PRIMARY parameter was outside the valid range.

System Action: The PRIMARY parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Qdisk ignored. " *value***" is not a valid volser.**

Severity: Completion.

Explanation: The value, *value*, specified for the QDISK parameter was not a valid volume serial name. The value specified for QDISK must be 6 characters or less.

System Action: The QDISK parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the QDISK parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Qdisk workarea allocation failed.

Severity: Completion.

Explanation: The allocation of the workarea for qdisk failed. number.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: qdiskp

200-RDW parameter not supported for the C server. RDW ignored.

Severity: Completion.

Explanation: The SITE command was entered with the RDW parameter, but the RDW parameter is not supported for the FTP C server.

System Action: The RDW parameter is ignored. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Recfm parameter *(value)* is invalid. Recfm ignored.

Severity: Completion.

Explanation: The value, *value*, specified for the record format is invalid. The record format must be one of: F, FA, FB, FBA, FBM, FBS, FM, V, VA, VB, VBA, VBM, VBS, VM$, VS, U, or blank.

System Action: The RECFM value is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the RECFM parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Replies are available in the following languages
200- *language*

Severity: Completion.

Explanation: A SITE QLANG command was issued to query the languages that are available for reply texts. *language* is the name of a language that can be requested for FTP replies by using the SITE LANG command. The "200-*language*" reply may appear multiple times if more than one language is available.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Representation type is Ascii NonPrint

Severity: Completion.

Explanation: The FTP server has received a valid TYPE subcommand, and the data representation type has been changed to ASCII NonPrint.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is Big5

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 8 command from the FTP client. The current transfer type has successfully been changed to Big5.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

200 Representation type is EBCDIC IBM kanji

Severity: Completion.

Explanation: The FTP TYPE F command has been accepted, and the data transfer type is the IBM kanji code, which is based on the EBCDIC code set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

200 Representation type is Ebcdic NonPrint

Severity: Completion.

Explanation: The FTP server has received a valid TYPE subcommand, and the data representation type has been changed to EBCDIC NonPrint.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is Hangeul

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 5 command from the FTP client. The current transfer type has successfully been changed to Hangeul DBCS.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

200 Representation type is IMAGE.

Severity: Completion.

Explanation: The FTP TYPE I command has been accepted, and the data transfer type is image (or binary). With the image transfer type, data is sent as continuous bits, packed into 8-bit bytes.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI EUC

Severity: Completion.

Explanation: The FTP server has received a valid TYPE subcommand, and the data representation type has been changed to Extended UNIX Code (EUC) KANJI.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1978

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 4 command from the FTP client. The current transfer type has successfully been changed to JIS 1978 Kanji DBCS, using the ASCII shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1978 shift-in ASCII

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 4 or TYPE B 4 A command from the FTP client. The current transfer type has successfully been changed to JIS 1978 Kanji DBCS, using the ASCII shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1978 shift-in JISROMAN

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 4 R command from the FTP client. The current transfer type has successfully been changed to JIS 1978 Kanji DBCS, using the JISROMAN shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1983

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 3 command from the FTP client. The current transfer type has successfully been changed to JIS 1983 Kanji DBCS, using the ASCII shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1983 shift-in ASCII

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 3 or TYPE B 3 A command from the FTP client. The current transfer type has successfully been changed to JIS 1983 Kanji DBCS, using the ASCII shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI JIS 1983 shift-in JISROMAN

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 3 R command from the FTP client. The current transfer type has successfully been changed to JIS 1983 Kanji DBCS, using the JISROMAN shift-in escape sequence.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KANJI Shift-JIS

Severity: Completion.

Explanation: The FTP server has received a valid TYPE subcommand, and the data representation type has been changed to KANJI Shift-JIS (Japanese Institute of Standards).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is KSC-5601

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 6 command from the FTP client. The current transfer type has successfully been changed to Korean Standard Code KSC-5601 DBCS.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is SChinese

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 9 command from the FTP client. The current transfer type has successfully been changed to Simplified Chinese.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Representation type is TChinese

Severity: Completion.

Explanation: The FTP server has received a valid TYPE B 7 command from the FTP client. The current transfer type has successfully been changed to Traditional Chinese (5550) DBCS.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Retpd parameter *(value)* **is not numeric. Retpd ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the RETPD parameter, but the value, *value*, specified for the RETPD parameter was not a numeric value. The value specified for the RETPD parameter must be a numeric value between 0 and 9999.

System Action: The RETPD parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Retpd parameter *(value)* **must be between 0 and 9999. Retpd ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the RETPD parameter, but the value, *value*, specified for the RETPD parameter was outside the valid range. The value of RETPD must be a numeric value between 0 and 9999.

System Action: The RETPD parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Sbdataconn invalid syntax. Specify SBDATACONN=(value1,value2) or SBDATACONN=dsn. SBDATACONN ignored.

Severity: Completion.

Explanation: A SITE subcommand has been entered with a SBDATACONN parameter that has incorrect syntax. The SBDATACONN parameter must be either a pair of code set names (the first must be an EBCDIC code set name, and the second must be an ASCII code set name), or the fully qualified name of an MVS data set or HFS file containing translate tables generated by the CONVXLAT utility.

System Action: FTP continues.

User or Operator Response: Resubmit the SITE subcommand with corrected syntax.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Sbdataconn parameter is too long. Maximum length for code page name is *length***. SBDATACONN ignored.**

Severity: Completion.

Explanation: A SITE subcommand has been entered with an SBDATACONN parameter specifying a invalid code page name. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Refer to the *OS/390 C/C++ Programming Guide* for information on supported code set converters and valid code set names. Resubmit the corrected SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Secondary parameter *(value)* **is not numeric. Secondary ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the SECONDARY parameter, but the value, *value*, specified for the SECONDARY parameter was not a numeric value.

System Action: The SECONDARY parameter is ignored.

User or Operator Response: Reissue the command with a valid value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Secondary parameter *(value)* **must be between 0 and 16777215. Secondary ignored.**

Severity: Completion.

Explanation: The SITE command was specified with the SECONDARY parameter, but the value, *value*, specified for the SECONDARY parameter was outside the valid range.

System Action: The SECONDARY parameter is ignored.

User or Operator Response: Reissue the command with a valid

value. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Site command was accepted

Severity: Completion.

Explanation: The SITE subcommand has been accepted and processed. If an error was detected in any SITE subcommand parameters, this reply will be preceded by one or more 200- replies indicating any parameter errors.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-SO/SI characters X'0E'/X'0F' used

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. The current transfer type now includes the use of SO/SI characters X'0E'/X'0F' to delimit ASCII DBCS data.

System Action: Program execution continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-SO/SI characters X'1E'/X'1F' used

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. The current transfer type now includes the use of SO/SI characters X'1E'/X'1F' to delimit ASCII DBCS data.

System Action: Program execution continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-SO/SI characters X'20'/X'20' used

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. The current transfer type now includes the use of SO/SI characters X'20'/X'20' (ASCII spaces) to delimit ASCII DBCS data.

System Action: Program execution continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-SQLCOL parameter ***(value)*** **is invalid. Sqlcol ignored.**

Severity: Completion.

Explanation: The value submitted for the SQLCOL parameter, which is used to determine the column headings of the output file, was not valid. Valid values for the SQLCOL parameter are:

Type	Description
Names	Uses the names of the DB2 SQL table columns. The labels are ignored.
Labels	Uses the labels of the SQL table columns. If any of the columns do not have labels, the corresponding column heading in the output file is given a heading of 'COLnnn'
Any	The label of the DB2 SQL table column is the first choice for column heading in the output file. If there is no label, the column name is used.

System Action: The SQLCOL parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the SQLCOL parameter. For more information about the SITE command, see the IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

200 Standard DBCS control used

Severity: Completion.

Explanation: The FTP server has received a valid DBCS TYPE command from the client. Standard DBCS control will be used. For JIS78KJ and JIS83KJ the selected escape sequences are used and for all other ASCII types no SO/SI characters are used. Data transfer is also set to mixed SBCS/DBCS.

System Action: Program execution continues. The current FTP transfer type is changed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

200-Storclass parameter ***(value)*** **is more than 8 characters. Storclass ignored.**

Severity: Completion.

Explanation: The value, *value*, specified for the Storclass parameter is not valid. The Storclass parameter value cannot be more than 8 characters long.

System Action: The Storclass parameter is ignored. FTP continues.

User or Operator Response: Resubmit the SITE command with a valid value for the STORCLASS parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200 Transferring PDS directory information.

Severity: Completion.

Explanation: PDS directory information will be transmitted along with the PDS member.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: sdir

200-Translate file ' *file*' **invalid. SBDATACONN ignored.**

Severity: Completion.

Explanation: A SITE subcommand has been entered with an SBDATACONN parameter that specified a filename, but FTP encountered an error while trying to load its translate tables using *file*. Possible errors include an invalid header record or incorrect file length. (The file must contain exactly 768 bytes of data.) The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Ensure that the filename specified is correct. It must be a fully qualified name, and the file must contain translate tables in the format generated by the CONVXLAT utility.

System Programmer Response: If FTP tracing was active, an EZY message will appear in FTP's system log and trace output explaining why FTP cannot use the file.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Translate file ' *file*' **not found. SBDATACONN ignored.**

Severity: Completion.

Explanation: A SITE subcommand has been entered with an SBDATACONN parameter that specified a filename, but FTP was unable open *file*. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Ensure that the filename specified on the SBDATACONN parameter is a fully qualified name of an existing MVS data set or HFS file.

System Programmer Response: If the filename is correct, look in the FTP server's trace for the reason why FTP cannot open the file.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Umask invalid syntax. Specify UMASK umask_value. UMASK ignored

Severity: Completion.

Explanation: The SITE command was issued with the UMASK parameter, but the syntax of the UMASK parameter was incorrect. The correct syntax is SITE UMASK *umask_value*, where *umask_value* is a 3 character octal number representing file permission bits.

System Action: The UMASK parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the UMASK parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Umask value invalid. Specify Umask value as 1 - 3 octal digits

Severity: Completion.

Explanation: The SITE command was issued with the UMASK parameter, but the value specified for the UMASK parameter was invalid. The value specified for UMASK should be a 1 to 3 character octal number representing file permission bits.

System Action: The UMASK parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the UMASK parameter. for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Umask value missing. Specify Umask value as 1 - 3 octal digits

Severity: Completion.

Explanation: The SITE command was issued with the UMASK parameter, but no value was specified for the UMASK parameter. The value specified for UMASK should be a 1 to 3 character octal number representing file permission bits.

System Action: The UMASK parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the UMASK parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Unable to set up conversion between *parm1* **and** *parm2***. SBDATACONN ignored.**

Severity: Completion.

Explanation: A SITE subcommand has been entered with an SBDATACONN parameter. A code set convertor was successfully opened, but FTP was unable to set up single-byte translate tables using the requested code sets. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Ensure that the requested code set names are for single-byte code pages. SBDATACONN is not supported for double-byte. Refer to the *OS/390 C/C++ Programming Guide* for information on supported code set converters and valid code set names.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Unit ignored. " *value*" is not a valid unit parameter

Severity: Completion.

Explanation: The value, *value*, specified for the UNIT parameter was not a valid unit name. The value specified for UNIT must be 8 characters or less.

System Action: The Unit parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the UNIT parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Unrecognized parameter '*parameter*=*operand*' on site command.

Severity: Completion.

Explanation: The SITE subcommand was entered with the parameter *parameter*, but *parameter* was not a valid SITE subcommand parameter.

System Action: The parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE subcommand with a valid parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Unrecognized parameter '*parameter*' on site command.

Severity: Completion.

Explanation: The SITE subcommand was entered with the parameter *parameter*, but *parameter* was not a valid SITE subcommand parameter.

System Action: The parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE subcommand with a valid parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

200-volume for " *dsname*" is not mounted and noautomount is specified. DCBDSN parameter ignored.

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, was on a volume which was not mounted to the system and the FTP server could not mount the volume because NOAUTOMOUNT had been specified at the server.

System Action: The DCBDSN parameter is ignored.

User or Operator Response: Issue the SITE subcommand with the AUTOMOUNT parameter to allow the volume to be mounted.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Volume ignored. " *value*" is not a valid volume parameter

Severity: Completion.

Explanation: The value, *value*, specified for the VOLUME parameter was not a valid volume serial. The value specified for VOLUME must be 6 characters or less.

System Action: The Volume parameter is ignored. FTP continues.

User or Operator Response: Reissue the SITE command with a valid value for the VOLUME parameter. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the SITE subcommand.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPMS

Procedure Name: site

200-Volume *volser* could not be found.

Severity: Completion.

Explanation: The QDISK parameter was entered with volume serial number *volser*. The requested volume serial number could not be found.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: qdiskp

200-Waiting for mount for " *dsname*"

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, is on a volume which is not mounted to the system. The volume must first be mounted to the system.

System Action: FTP waits for the volume for *dsname* to be mounted, then continues processing the DCBDSN parameter of the SITE subcommand.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-Waiting for recall of " *dsname*"

Severity: Completion.

Explanation: The SITE subcommand has been issued with the DCBDSN parameter. The data set specified as the model DCB, *dsname*, is migrated and must be recalled.

System Action: FTP waits for *dsname* to be recalled, then continues processing the DCBDSN parameter of the SITE subcommand.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: check_dcbds

200-XLATE parameter not supported for the C server. XLATE ignored.

Severity: Completion.

Explanation: The SITE command was entered with the XLATE parameter, but the XLATE parameter is not supported for the OE FTP C server.

System Action: The XLATE parameter is ignored. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMS

Procedure Name: site

---

202 ACCT command not required - ignored.

Severity: Completion.

Explanation: The FTP server received the ACCT command, which is not supported under MVS. The command is ignored.

System Action: FTP continues.

User or Operator Response: Use the USER command instead of the ACCT command. For more information about the USER command, see the *OS/390 TCP/IP OpenEdition User's Guide*, or type HELP USER at the command line.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: nosyntchk_cmd

---

202 ALLO not necessary, you may proceed

Severity: Completion.

Explanation: The FTP server received the ALLO subcommand, which is not supported under MVS. The command is ignored.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: nosyntchk_cmd

---

202 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

202 SITE not necessary: you may proceed

Severity: Completion.

Explanation: The SITE subcommand was entered with no parameters.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: site_cmd

---

211 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

213 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

213 Status: *number* **bytes transferred**

Severity: Completion.

Explanation: The STAT command was received as an Out Of Band command during data transfer. When the STAT command is received during data transfer, the server returns only the number of bytes transferred on the data connection.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHC

Procedure Name: server_oob

214 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

215 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

215 MVS is the operating system of this server. FTP Server is the C-server running on OE.

Severity: Completion.

Explanation: The FTP server handling the file transfer uses an MVS operating system. This is the OE FTP server which runs under OMVS and is capable of handling HFS files.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: noarg_cmd

---

220 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

220 Connection will close if idle for more than *timeout*

Severity: Completion.

Explanation: The FTP server was started with a timeout value of *timeout*. If there is no activity on the control connection between the client and the server for the specified amount of time, the control connection will be terminated.

Note: The inactivity timer is not in effect during data transfer. The inactivity timer is only in effect during the time between the successful completion of one subcommand and the issuing of the next subcommand.

Note: The inactivity timer is for the control connection only. It has no effect on the data connection.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: spawn_ftps

---

220 Connection will not timeout.

Severity: Completion.

Explanation: The FTP server was started with no control connection timeout value. The control connection between the client and the server will stay active indefinitely, regardless of whether or not there is activity on the connection.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: spawn_ftps

220- *jobname* **IBM MVS** *version_release* **at** *hostname***,** *init_time* **on** *init_date***.**

Severity: Completion.

Explanation: The FTP client has successfully contacted the FTP server. The FTP server is at host *hostname*. The FTP server job, *jobname*, was initiated at *init_time* on *init_date*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: spawn_ftps

---

220 Reinitialized; Ready for new user.

Severity: Completion.

Explanation: The FTP server is reinitialized and ready for a new user.

System Action: The FTP server is reinitialized and waits for a new user.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRM

Procedure Name: rein

221 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

221 Quit command received. Goodbye.

Severity: Completion.

Explanation: The Quit command has been received. The FTP session ends and the connection to the host is terminated.

System Action: The FTP session ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRM

Procedure Name: quit

226 Abort successful.

Severity: Completion.

Explanation: The ABOR command was received and processed successfully.

System Action: If a command is in progress, the command in progress is terminated. If a data transfer is in progress, the data transfer is terminated and the data connection is closed. If there is no command in progress no action is taken. The control connection remains active and the FTP server waits for the next command from the client.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHC, EZAFTPRM

Procedure Name: serve_oob, abor

226 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

227 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

227 Entering Passive Mode *(h1,h2,h3,h4,p1,p2)*

Severity: Completion.

Explanation: The server has received the PASV subcommand. The server has successfully entered passive mode and is listening on the socket indicated by *h1,h2,h3,h4,p1,p2*, where *h1-h4* are the host ip address and *p1,p2* are the port number.

System Action: FTP creates the server socket and listens on that socket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

Procedure Name: pasv

230 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

230-Password has been changed.

Severity: Completion.

Explanation: The PASS command was entered in the format PASS old_pass/new_pass/new_pass to change the password during login. The user's password has been changed. The user can no longer log on to the system using the old password.

System Action: The user's password is changed. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

230 *user_id* **is logged on. Working directory is "***directory***".**

Severity: Completion.

Explanation: The named user has successfully logged on.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

250 Cancel Successful

Severity: Completion.

Explanation: A job submitted through the Job Entry Subsystem (JES) has been canceled due to user request.

System Action: The job is canceled. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

250 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

250-Compression reduced bytes transferred by *amount* **percent**

Severity: Completion.

Explanation: Data is being transferred while in mode c (compressed). The compression algorithm has reduced by *amount* percent the number of bytes of data that are transferred for the request.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesGet, jesPut, retrieve, store_data

---

250 *dsname* **deleted**

Severity: Completion.

Explanation: The indicated data set was deleted by user request.

System Action: The data set is deleted. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD, EZAFTPRF

Procedure Name: rmd, delete

---

250 *dsname* **deleted from migration volume.**

Severity: Completion.

Explanation: The data set is migrated and has been deleted from the migration volume. Note that if the migration and recall facility is currently inactive, the actual delete may occur later when the facility is activated.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

---

250 Data structure is File.

Severity: Completion.

Explanation: The STRU subcommand has been received by the FTP server requesting that the file structure for data transfer be set to FILE. The server has processed the command and changed the file structure to FILE.

System Action: The File structure is changed to FILE structure.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: stru

---

250 Data structure is Record.

Severity: Completion.

Explanation: The STRU subcommand has been received by the FTP server requesting that the file structure for data transfer be set to RECORD. The server has processed the command and changed the file structure to RECORD.

System Action: The file structure is set to RECORD structure

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: stru

---

250 " *directory***" is the working directory name prefix.**

Severity: Completion.

Explanation: A CWD command was issued which caused the current working directory of the server to be changed to the MVS high level qualifier *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

---

250 " *directory*" **partitioned data set is working directory**

Severity: Completion.

Explanation: A CWD command was issued which caused the current working directory of the server to be changed to the MVS partitioned data set *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

---

250 directory remains " *old_directory*"**.**

Severity: Completion.

Explanation: A CWD subcommand was entered to change the current working directory, but the FTP server was unable to successfully change the directory. The directory remains unchanged from the old directory.

System Action: The directory is unchanged. FTP continues.

User or Operator Response: This reply is accompanied by other replies which indicate the problem. Correct the errors indicated by the other replies.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

250-Error mounting " *dsname*"**;**

Severity: Completion.

Explanation: A CWD command was issued to change the current working directory to *dsname*. *dsname* was an existing MVS data set which was catalogued on an unmounted volume. The FTP server needed to mount the volume for the data set in order to determine whether or not the data set was a partitioned data set. The mount of the volume for *dsname* failed. The server was unable to complete the CWD command.

System Action: The server attempts to mount the data set, but the mount is unsuccessful. The current working directory is not changed.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why the mount of the volume failed and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: cwd2.

250-Error retrieving " *dsname*"

Explanation: A CWD command was issued to change the current working directory to *dsname data set* was an existing MVS data set which was migrated. The FTP server needed to recall the data set in order to determine whether or not the data set was a partitioned data set. The recall of migrated data set *dsname* failed.

System Action: The server attempts to recall the data set, but the recall is unsuccessful. The current working directory is not changed.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why the recall of the migrated data set failed and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: cwd2.

---

250 HFS directory *directory* **is the current working directory**

Severity: Completion.

Explanation: The user has issued the CWD command to change the current working directory. The CWD command has completed successfully and has changed the current working directory to the HFS directory indicated by *directory*.

System Action: The current working directory is changed to the new directory.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

250-It is known to JES as *job_ID***.**

Severity: Completion.

Explanation: This reply displays the job ID assigned to a job by the Job Entry System (JES). Use this job ID to access the job through the JES.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

---

250 List completed successfully.

Severity: Completion.

Explanation: A list command has been successfully completed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD, EZAFTPMJ

Procedure Name: list2, jesStatus

---

250 Nlst completed successfully

Severity: Completion.

Explanation: A command was issued to obtain a NLST. The NLST command completed successfully.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesStatus

250 *old data set* **renamed to** *new data set*

Severity: Completion.

Explanation: The indicated data set has been renamed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: hfs_rnto, rnto

---

250 Transfer completed (data was truncated)

Severity: Completion.

Explanation: A store request or a request to submit a job contains data whose record length is longer than that of the destination data set. Some of the data was truncated (lost).

System Action: FTP continues.

User or Operator Response: Specify a logical record length for the destination data set that is big enough for the largest record that is transferred.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRS

Procedure Name: jesPut, store_data

---

250 Transfer completed (lines were wrapped)

Severity: Completion.

Explanation: A store request or a request to submit a job contains data whose record length is longer than that of the destination data set. However, since the WrapRecord option is in effect, the data is wrapped into the next record instead of being truncated (lost).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRS

Procedure Name: jesPut, store_data

---

250 Transfer completed *(number translation errors)*

Severity: Completion.

Explanation: A data transfer request (retrieve or store) or a request to submit a job contains data that must the translated to or from a double-type character set. A total of *number* translation errors occurred during the translation process.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesPut, retrieve, store_data

---

250 Transfer completed successfully.

Severity: Completion.

Explanation: The requested data transfer command has successfully completed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesGet, jesPut, retrieve, store_data, store_null_dir

---

250 Transfer completed successfully, warning SELECT file too large

Severity: Completion.

Explanation: A SQL query was submitted through FTP. The length of the SQL statement in the input file exceeded 32765 characters. The SQL statement was truncated to 32765 characters and submitted to DB2.

System Action: FTP continues.

User or Operator Response: Check the input file used to submit the query. If necessary, correct the SQL statement in the input file, and resubmit the query.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

250-Waiting for mount for " *dsname***"**

Severity: Completion.

Explanation: A CWD command was issued to change the current working directory to *dsname* *data set* was an existing MVS data set which was catalogued on an unmounted volume. The FTP server needed to mount the volume for the data set in order to determine whether or not the data set was a partitioned data set.

System Action: The FTP server waits for the volume to be mounted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2.

250-Waiting for recall of " *dsname***"**

Severity: Completion.

Explanation: A CWD command was issued to change the current working directory to *dsname*. *dsname* was an existing MVS data set which was migrated. The FTP server needed to recall the data set in order to determine whether or not the data set was a partitioned data set.

System Action: The server attempts to recall the data set.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2.

250-Working Directory for PUT is NULL Device;
250 for GET is " *directory***" partitioned data set.**

Severity: Completion.

Explanation: The user has entered the CWD *DEV.NULL command to change the current working directory of the server to the NULL directory. When the current working directory is the NULL directory, data transferred to the server from the client (STOR or STOU) will not be stored at the server, but will be received and discarded. The NULL directory is used only for data transferred from the client to the server. Data transferred from the server to the client (RETR) will continue to use the working directory which was in effect prior to the issuance of the CWD *DEV.NULL command. The directory which will be used for the RETR command is the partitioned data set *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

250-Working Directory for PUT is NULL Device;
250 for GET is HFS directory *directory*

Severity: Completion.

Explanation: The user has entered the CWD *DEV.NULL command to change the current working directory of the server to the NULL directory. When the current working directory is the NULL directory, data transferred to the server from the client (STOR or STOU) will not be stored at the server, but will be received and discarded. The NULL directory is used only for data transferred from the client to the server. Data transferred from the server to the client (RETR) will continue to use the working directory which was in effect prior to the issuance of the CWD *DEV.NULL command. The directory which will be used for the RETR command is the HFS directory *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

250-Working Directory for PUT is NULL Device;
250 for GET is " *directory*"**name prefix.**

Severity: Completion.

Explanation: The user has entered the CWD *DEV.NULL command to change the current working directory of the server to the NULL directory. When the current working directory is the NULL directory, data transferred to the server from the client (STOR or STOU) will not be stored at the server, but will be received and discarded. The NULL directory is used only for data transferred from the client to the server. Data transferred from the server to the client (RETR) will continue to use the working directory which was in effect prior to the issuance of the CWD *DEV.NULL command. The directory which will be used for the RETR command is the MVS high level qualifier *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

---

257 command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

257 " *directory*"**created.**

Severity: Completion.

Explanation: The indicated directory has been created.

System Action: The directory is created. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

257 " *directory*"**is the HFS working directory.**

Severity: Completion.

Explanation: The current working directory for storing and retrieving data from the FTP server is HFS directory *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

---

257 " *directory*"**is working directory.**

Severity: Completion.

Explanation: The current working directory for storing and retrieving data from the FTP server is MVS high level qualifier *directory*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

257 "' *directory*" partitioned data set is working directory

Severity: Completion.

Explanation: If the current working directory is a partitioned data set (PDS), then the user's PDS name is listed here. For more information about PDSs, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

---

257 MKD failed. Error mounting volume.

Severity: Completion.

Explanation: The MKD command was issued to create the new PDS but the server was unable to mount the volume on which the data set is to be allocated.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Contact the system programmer for the server system.

System Programmer Response: Determine why the volume could not be mounted and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: mkd

257-Waiting for volume mount for volume *volume*

Severity: Completion.

Explanation: The FTP server must wait for the indicated volume to be mounted before the command can be processed.

System Action: The volume is mounted and the command is processed. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMV

Procedure Name: seq_create_file

257-Working Directory for PUT is NULL Device;
257 for GET is " *directory*" **partitioned data set.**

Severity: Completion.

Explanation: The current working directory for data transferred to the server from the client (STOR or STOU) is the NULL directory. Data transferred to the server will be received and discarded rather than stored at the server's host system. MVS Partitioned Data Set *directory* will be used as the current working directory for transferring data from the server to the client (RETR command).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

257-Working Directory for PUT is NULL Device;
257 for GET is HFS directory *directory*

Severity: Completion.

Explanation: The current working directory for data transferred to the server from the client (STOR or STOU) is the NULL directory. Data transferred to the server will be received and discarded rather than stored at the server's host system. HFS directory *directory* will be used as the current working directory for transferring data from the server to the client (RETR command).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

257-Working Directory for PUT is NULL Device;
257 for GET is " *directory*" **name prefix.**

Severity: Completion.

Explanation: The current working directory for data transferred to the server from the client (STOR or STOU) is the NULL directory. Data transferred to the server will be received and discarded rather than stored at the server's host system. MVS high level qualifier *directory* will be used as the current working directory for transferring data from the server to the client (RETR command).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: pwd

---

300 command terminated due to server shutdown in progress

Severity: Intermediate.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

300 Restart command accepted, parameter=*restart checkpoint*

Severity: Intermediate.

Explanation: If a checkpoint for the restart is found by FTP, this reply indicates that the restart is successful. The value for the checkpoint parameter represents the ordinal (number) of the data byte where the restart begins.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rest

---

331 command terminated due to server shutdown in progress

Severity: Intermediate.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

331 Send password please.

Severity: Intermediate.

Explanation: The USER command has been entered to begin user login processing. The FTP server requires a password to validate the user id.

System Action: FTP continues.

User or Operator Response: Use the PASS subcommand in FTP to supply your password to the remote host. For more information, see *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: user

---

350 command terminated due to server shutdown in progress

Severity: Intermediate.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

350 RNFR accepted. Please supply new name for RNTO.

Severity: Intermediate.

Explanation: The RNFR command, which specifies the data set to be renamed, has been accepted. The FTP server is prompting the user to supply the new name for the data set.

System Action: FTP continues.

User or Operator Response: Use the RNTO command to specify the new name for the data set to be renamed. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the RNFR and RNTO subcommands, or type HELP RNTO at the command line.

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPRF

Procedure Name: hfs_renamefr, renamefr

---

400 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

421 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

421 connection terminated - seteuid failure during *cmd* command processing.

Severity: Transient negative completion.

Explanation: The client has entered either the USER command to change the user_id of the FTP session, or the REIN command to reinitialize an FTP session back to the state of a new connection which has not logged on a user_id. In order to reset the user_id to a new user_id or to a state of no user_id, the server must change the user_id associated with the server to a “superuser” user_id. The server was unable to return to the “superuser” environment.

System Action: The FTP session is terminated.

User or Operator Response: Retry the connection and the failing command. If the problem persists, contact the system programmer.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The seteuid error will create a trace record with a return code. Determine the cause of the error and correct the problem.

Source Data Set: EZAFTPRA, EZAFTPRM

Procedure Name: user, rein

421 Open rejected due to insufficient resources.

Severity: Transient negative completion.

Explanation: The FTP server attempted to create a new address space for the client session, but was unable to do so.

System Action: FTP continues. The client connection is terminated.

User or Operator Response: If the problem persists, contact the system programmer.

System Programmer Response: If necessary, recreate the problem with FTP C server traces. The trace messages will show the exact error received by the server when it attempted to fork the new address space. Correct the error indicated by the traces.

Source Data Set: EZAFTPSK

Procedure Name: spawn_ftps

---

421 User Exit rejects open for connection

Severity: Transient negative completion.

Explanation: The FTP server is running with the security user exit FTCHKIP. The user exit does not allow the IP address to access the FTP server.

System Action: FTP continues. The client connection is rejected.

User or Operator Response: Contact the owner of the FTP server to have the IP address authorized in the FTCHKIP user exit.

System Programmer Response: If necessary, update the FTCHKIP user exit to allow the ip address to access the server.

Source Data Set: EZAFTPSK

Procedure Name: accept_client

---

425 Can't open data connection.

Severity: Transient negative completion.

Explanation: The server cannot open the data connection to transfer the data set.

System Action: The data set is not transferred. FTP continues.

User or Operator Response: None.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The data connection error will create a trace record with a socket function return code. Determine the cause of the socket error and correct the problem.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesGet, jesPut, jesStatus, retrieve, store_data, store_null_dir

---

425 Can't open passive connection.

Severity: Transient negative completion.

Explanation: The FTP server has received a PASV command requesting that the server open a passive data connection (server does the “listen” rather than the “connect”). The server was unable to open the passive connection.

System Action: The PASV command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The open data connection error will create a trace record with a socket function return code. Determine the cause of the socket error and correct the problem.

Source Data Set: EZAFTPSK

Procedure Name: pasv

---

425 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

425-Severe error during data connection.
425-Userid info lost, please re-issue USER command.

Severity: Transient negative completion.

Explanation: The FTP server was attempting to open a data connection to transfer data. In order to open a data connection, the FTP server must return to “superuser” to bind the socket, then return to the userid of the client. At some point in this processing, the server encountered an error and was unable to return to the userid of the client.

System Action: The data transfer is terminated. The client session is re-initialized back to a point of a new session with no user logged in.

User or Operator Response: Reissue the USER and PASS subcommands to log back on to the desired userid. Retry the data transfer. If the problem persists, contact the system programmer.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The seteuid error will create a trace record with a return code. Determine the cause of the seteuid error and correct the problem. If the error message in the traces indicates an “internal error ”, contact the IBM Support Center.

Source Data Set: EZAFTPSK

Procedure Name: data_connect

---

425 Unable to open data connection.

Severity: Transient negative completion.

Explanation: The server cannot open the data connection to transfer the data.

System Action: The data is not transferred. FTP continues.

User or Operator Response: None.

System Programmer Response: If necessary, re-create the problem with FTP server trace active. The data connection error will create a trace record with a socket function return code. Determine the cause of the socket error and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: list2

426 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

426 Connection closed; transfer aborted.

Severity: Completion.

Explanation: The FTP server received an Out Of Band ABOR subcommand from the FTP client requesting that the data transfer in progress be terminated. The FTP server has aborted the data transfer.

System Action: The data transfer in progress is aborted. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHC

Procedure Name: serve_oob

---

426 Data connection aborted

Severity: Transient negative completion.

Explanation: The FTP server closed the data connection because of an error reported in a previous reply.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

450 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

450 JES not responding, time limit exceeded

Severity: Transient negative completion.

Explanation: The FTP address space could not obtain the required lock within the required time.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

450 JESGET failed, could not allocate send buffer

Severity: Transient negative completion.

Explanation: While attempting to send a file, a buffer was not available.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None

Source Data Set: EZAFTPMJ

Procedure Name: jesRetrieveSpoolFile

450-Record too long

Severity: Transient negative completion.

Explanation: During the transmission of a file, an incomplete record was encountered.

System Action: FTP continues.

User or Operator Response: Re-issue the command.

System Programmer Response: None

Source Data Set: EZAFTPMJ

Procedure Name: jesWrtStream

---

451 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

451-Compression reduced bytes transferred by *amount* percent

Severity: Completion.

Explanation: Data is being transferred while in mode c (compressed). The compression algorithm has reduced by *amount* percent the number of bytes of data that are transferred for the request.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

451-Data set is full (19).

Severity: Completion.

Explanation: A store request has failed because the destination data set is full. The C runtime errno code of 19 indicates that an unrecoverable error has permanently marked the data set in error.

System Action: FTP continues.

User or Operator Response: Request a larger space allocation for the destination data set and try the transfer again.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

451-Data set is full (33).

Severity: Completion.

Explanation: A store request has failed because the destination data set is full. The C runtime errno code of 33 says that an attempt was made to extend a non-extendable file.

System Action: FTP continues.

User or Operator Response: Request a larger space allocation for the destination data set and try the transfer again.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

451-Data set is full (65).

Severity: Completion.

Explanation: A store request has failed because the destination data set is full. The C runtime errno code of 65 says that a write system error occurred.

System Action: FTP continues.

User or Operator Response: Request a larger space allocation for the destination data set and try the transfer again.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

451 data transfer aborted - command terminated due to server shutdown in progress

Severity: Completion.

Explanation: The FTP server was in the process of transferring data when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: Data transfer is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

451 Dele failed due to internal error

Severity: Transient negative completion.

Explanation: A required data area was not available for the requested JES function to complete.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

451 Dele failed, too many entries

Severity: Transient negative completion.

Explanation: A request was made to JES to delete JOBS. To many jobs exist for JES to process this request.

System Action: FTP continues.

User or Operator Response: Re-issue the command. If the problem persists, request assistance from your system support personnel.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

451 Error *(rc)* **closing data set** *dsname*

Severity: Transient negative completion.

Explanation: The close of the data set was unsuccessful.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Resubmit the command.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

451 Index *index* **is greater than number of spool files for** *jobid*

Severity: Transient negative completion.

Explanation: A specific spool file was requested to be retrieved from JES using the FTP GET command. JES output files are numbered, and the DATASET_NUMBER specified for the spool file desired was not found for this job.

System Action: FTP continues.

User or Operator Response: Re-issue the FTP GET command with the correct JOBID and DATASET_NUMBER.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

451 Internal storage allocation error, JesPutGet aborted

Severity: Transient negative completion.

Explanation: While attempting to retrieve an output file for a JOB, an insufficient storage condition existed.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

451 Internal storage allocation error, try again later

Severity: Transient negative completion.

Explanation: While attempting to retrieve an output file for a JOB, an insufficient storage condition existed.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

451 List failed due to internal error

Severity: Transient negative completion.

Explanation: A required data area was not available at the time the command was issued.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesStatus

---

451 List failed, too many entries

Severity: Transient negative completion.

Explanation: The request resulted in more data than the data area could contain.

System Action: FTP continues.

User or Operator Response: Re-issue the command later. If the problem persists, request assistance from system support personnel.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesStatus

---

451 Nlst failed due to internal error

Severity: Transient negative completion.

Explanation: A required data area was not available at the time the command was issued.

System Action: FTP continues.

User or Operator Response: Re-issue the command later.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesSTatus

---

451 Nlst failed, too many entries

Severity: Transient negative completion.

Explanation: The request resulted in more data than the data area could contain.

System Action: FTP continues.

User or Operator Response: Re-issue the command later. If the problem persists, request assistance from system support personnel.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesStatus

451-Record received was too short (4).

Severity: Completion.

Explanation: A store request has failed because one of the records that is transferred is too short for the destination data set. This error occurs when the destination data set is a fixed record format, the transfer mode is b (blocked) or c (compressed), and a short record is transferred. The C runtime errno code of 4 says that a specified record length is not large enough.

System Action: FTP continues.

User or Operator Response: Ensure that each fixed format record has the correct length or choose a non-fixed record format (e.g., variable).

System Programmer Response:

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

451 Renaming attempt failed.

Severity: Transient negative completion.

Explanation: A RNTO command was issued to rename a member of a partitioned data set. The rename was unsuccessful because a system resource was not available.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: None.

System Programmer Response: If possible, re-create the problem with FTP server trace active and contact the IBM Support Center with the results.

Source Data Set: EZAFTPRF

Procedure Name: rnto

---

451 Transfer aborted due to file error. *file_status*.

Severity: Transient negative completion.

Explanation: The transfer of data ended because of an error writing to the data set. For new MVS data sets that were being written, the *file_status* is either *File is deleted* or *File is catalogued*. The status is affected by the current value of the CONDDISP parameter (see the SITE command) and whether the checkpoint was in progress.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Get a trace of the store operation and contact the IBM Software Support Center.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

451 Transfer aborted due to receive error. *file_status*.

Severity: Transient negative completion.

Explanation: The transfer of data ended because of an error receiving data from the data connection. For new MVS data sets that were being written, the *file_status* is either *File is deleted* or *File is catalogued*. The status is affected by the current value of the CONDDISP parameter (see the SITE command) and whether or not checkpoint was in progress.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Get a trace of the store operation and contact the IBM Software Support Center.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

451 Transfer aborted: file error.

Severity: Transient negative completion.

Explanation: The transfer of data ended because of an error reading the data set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Get a trace of the retrieve operation and contact the IBM Software Support Center.

Source Data Set: EZAFTPMJ, EZAFTPRR

Procedure Name: jes_submit_job, JesRetrieveSpoolFile, retrieve

---

451 Transfer aborted: internal error while processing SQL request

Severity: Transient negative completion.

Explanation: FTP has encountered an internal error while processing a SQL request. No data is sent.

System Action: FTP continues.

User or Operator Response: Try again. If the problem persists, contact the system programmer for FTP.

System Programmer Response: If the problem occurs repeatedly, get a trace of the FTP operation and contact the IBM Support Center.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

451 Transfer aborted: send error

Severity: Transient negative completion.

Explanation: The server was attempting to send data to the client over the data connection, but an error occurred executing the send() socket function for the data connection. The data transfer is ended.

System Action: Control is returned to the client connection for further command processing.

User or Operator Response: The command can be retried. If the problem persists, contact the system programmer

System Programmer Response: If necessary, re-create the problem with FTP server traces active. The send() error will create a trace record with the send() return code. Determine the cause of the send error and correct the problem.

Source Data Set: EZAFTPMJ, EZAFTPRD, EZAFTPRR

Procedure Name: jes_submit_job, jesRetrieveSpoolFile, list2, retrieve

---

451 Transfer aborted: SQL FETCH error *(code)*

Severity: Transient negative completion.

Explanation: A SQL query was submitted through FTP, but DB2 has encountered an error retrieving the data. Data retrieval for this query is discontinued. Data in the output file may be incomplete. *code* indicates the reason for the failure.

System Action: FTP continues.

User or Operator Response: Resubmit the query. If the problem persists, contact the system programmer for FTP.

System Programmer Response: See "SQL Return Codes" in the *DB2 Messages and Codes* for an explanation of *code* and correct the problem.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

451 Transfer completed

Severity: Completion.

Explanation: A job has been successfully submitted,

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

---

451 Unable to acquire JES access, JesPutGet aborted

Severity: Transient negative completion.

Explanation: The FTP address space could not obtain the required FTP/JES lock within the required time.

System Action: FTP continues.

User or Operator Response: Re-issue the command later

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

451 Unable to acquire JES access, try again later

Severity: Transient negative completion.

Explanation: The FTP address space could not obtain the required FTP/JES lock within the required time.

System Action: FTP continues.

User or Operator Response: Re-issue the command later

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

451-Unable to get Jobid

Severity: Transient negative completion.

Explanation: A job was submitted to JES. No Jobid was returned after the completion of the submit.

System Action: FTP continues.

User or Operator Response: Check the JCL file for an error in the JOBCARD.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPut

---

452 command terminated due to server shutdown in progress

Severity: Transient negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

452 Insufficient storage to process site command.

Severity: Transient negative completion.

Explanation: The SITE command was entered, but there was not enough storage available for the server to parse the site command.

System Action: FTP continues.

User or Operator Response: Try the command at a later time when more storage may be available.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: site_cmd

---

452 Transfer aborted: insufficient storage to process SQL request

Severity: Transient negative completion.

Explanation: A SQL query was being attempted through FTP, but there is not enough system storage available for FTP to compete the request.

System Action: No data is sent. FTP continues.

User or Operator Response: Try again. If the problem persists, contact the system programmer for the FTP server.

System Programmer Response: Determine why FTP is unable to acquire dynamic storage and correct the problem.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

500 command failed - command too long

Severity: Permanent negative completion.

Explanation: The FTP server attempted to receive a command from the client connection, but the command received was larger than the FTP server command buffer. The maximum allowed command length is 1099 bytes, including control characters.

System Action: The command is rejected. Control returns to the client connection for further command processing.

User or Operator Response: If possible, break the command into multiple, smaller, commands (for example, a SITE command with multiple parameters can be broken into several SITE commands with fewer parameters). If it is not possible to break the command into smaller pieces, contact the IBM $ Software Support Center with requirements for a larger command buffer.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

500 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

500 invalid command *command*

Severity: Permanent negative completion.

Explanation: The client connection entered the command *command*, which the FTP server did not recognize as a valid FTP command. The command is rejected.

System Action: Control is returned to the client for further command processing.

User or Operator Response: Issue a valid FTP command.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

500 Invalid restart parameter.

Severity: Permanent negative completion.

Explanation: You attempted to restart a checkpointed data transfer command and the restart marker that was received at the server is incorrect.

System Action: The data transfer command is not restarted. FTP continues.

User or Operator Response: Make sure that the *hlq*.FTP.CHECKPOINT data set is valid. This data set was created by the client during a checkpointed command and must not be altered after the checkpointed command is unsuccessful and before the restart is entered. Failure to observe this procedure will affect the restart.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rest

500 unknown command *command*

Severity: Permanent negative completion.

Explanation: The client connection entered the command *command*, which the FTP server did not recognize as a valid FTP command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Issue a valid FTP command.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

500 User Exit denies Userid ' *user_ID*' from using Command '*command*'.

Severity: Permanent negative completion.

Explanation: The client logged onto the server as user ID *user_ID* and attempted to issue the FTP command *command*. The FTP server was running with the user-written exit routine FTCHKCMD, and the FTCHKCMD user exit prevented the user ID *user_ID* from issuing command *command*.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer for the server system.

System Programmer Response: If necessary, change the user exit FTCHKCMD to allow the user to issue the command. The client must terminate the FTP session and then reconnect to the server in order to pick up changes to the user exit routines.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

501 * is not last char

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with the * global file-name character in the path name, but the * was not the last character in the qualifier. For MVS FTP, the * global file-name character can only appear at the end of any qualifier in the data set name.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Enter the command again with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 A qualifier in " *pathname*" begins with an invalid character

Severity: Permanent negative completion.

Explanation: The pathname in the subcommand entered specified an MVS data set, but one of the qualifiers in the data set name began with an invalid character. Qualifiers in an MVS data set name can only begin with a letter, a '$', an '@', or a '#'.

System Action: The subcommand is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the subcommand with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: validate_mvsname

---

501 A qualifier in " *pathname*" is more than 8 characters

Severity: Permanent negative completion.

Explanation: The pathname in the subcommand entered specified an MVS data set, but one of the qualifiers in the data set name was longer than 8 characters. Qualifiers in an MVS data set name cannot exceed 8 characters.

System Action: The subcommand is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the subcommand with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: validate_mvsname

---

501 A qualifier in " *pathname*" contains an invalid character

Severity: Permanent negative completion.

Explanation: The pathname in the subcommand entered specified an MVS data set, but one of the qualifiers in the data set name contained an invalid character. Qualifiers in an MVS data set name can only contain letters, numbers, a '$', an '@', '#', '-', or a 'x'C0''.

System Action: The subcommand is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the subcommand with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRU

Procedure Name: validate_mvsname

---

501 command aborted - error in local processing

Severity: Permanent negative completion.

Explanation: An internal programming error occurred that caused the parser to end up within the routines that process the valid FTP commands while processing an incorrect command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the FTP server system.

System Programmer Response: Contact the IBM Software Support Center. The FTP server output should provide additional information regarding the failure.

Source Data Set: EZAFTPPA

Procedure Name: noarg_cmd, nosyntchk_cmd, onearg_cmd

---

501 *command* **command syntax error : too many parameters**

Severity: Permanent negative completion.

Explanation: The client issued the FTP command *command*, but the command contained more parameters than allowed for that command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the command with the correct number of parameters.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: noarg_cmd, type_cmd

---

501 *command* **supported only in Stream mode**

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered, but the FTP session was in a mode other than stream mode. The LIST and NLST command are supported only in stream mode.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Issue the MODE S command to change the mode to stream mode, then reissue the LIST or NLST command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

501 CWD ../ only valid for HFS directories

Severity: Permanent negative completion.

Explanation: The CWD command was entered with a pathname of ../ to back up one directory level, but the current working directory was either an MVS partitioned data set, or an MVS high level qualifier. The “../” notation is only valid when the current working directory is an HFS directory.

System Action: The CWD command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the CWD command with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

---

501 Data set name too long. Use MVS naming conventions

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with a data set name that was longer than the MVS maximum of 44 characters.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the LIST or NLST command with the correct data set name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 Data set name too long. Use MVS or HFS naming conventions.

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with a data set name that was longer than the MVS maximum of 44 characters or the HFS maximum of 1023 characters.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the LIST or NLST command with the correct data set or HFS file name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 ‘ *data_set(member)*’ **requests members but** *data_set* **is not a partitioned data set.**

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered using the MVS syntax *data_set(member)*, which indicates that members of the PDS *data_set* are to be listed; however, *data_set* is not a partitioned data set.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Verify that the correct data set name was entered. Reissue the LIST or NLST command with the correct data set name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 Directory is already NULL

Severity: Permanent negative completion.

Explanation: The CDUP, or CD .. command was issued to back up the directory name one level, but the directory name was already back to the NULL directory ("").

System Action: The CDUP command is rejected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

501 Error - existing kanji type invalid

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is for a DBCS data type, but an internal error has occurred and the current data type is no longer valid.

System Action: The FTP connection with the client is terminated.

User or Operator Response: Reconnect to the FTP server and issue the desired TYPE command. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide. If this reply is received repeatedly, make a note of the last few FTP commands you have issued that cause this result, and contact your system programmer

System Programmer Response: Contact the IBM Support Center with output from the FTP server trace, if available.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Error - existing type invalid

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. FTP tried to continue with the current data transfer type, but an internal error has occurred and the current data type is no longer valid.

System Action: The FTP connection with the client is terminated.

User or Operator Response: Reconnect to the FTP server and issue the desired TYPE command. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide. If this reply is received repeatedly, make a note of the last few FTP commands you have issued that cause this result, and contact your system programmer

System Programmer Response: Contact the IBM Support Center with output from the FTP server trace, if available.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 File name too long. Use HFS naming conventions.

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with a file name that was longer than the HFS maximum of 1023 characters.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the LIST or NLST command with the correct HFS file name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 Invalid data set name *"dsname"*. Use MVS Dsname conventions.

Severity: Permanent negative completion.

Explanation: The data set name violates one of the MVS file naming conventions and cannot be used to reference a data set at the server.

System Action: The data set is not sent. FTP continues.

User or Operator Response: Rename the data set in compliance with MVS data set naming conventions. For more information about MVS data set naming conventions, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: delete, renamefr, rnto, retrieve, store_data

---

501 Invalid data set name. Use MVS Dsname conventions.

Severity: Permanent negative completion.

Explanation: The command was entered with a data set name, but the data set name did not meet MVS data set name conventions. The data set name must:

- Be no more than 44 characters total
- Each qualifier must be no more than 8 characters total
- Each qualifier must start with a letter or with a $, @, or #
- The 2nd-8th characters of each qualifier must be either a letter, a number, or a $, @, #, -, or }.

System Action: The command is rejected.

User or Operator Response: Reissue the command with the corrected data set name.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesGet, jesPut, retrieve, store_data

---

501 Invalid directory name - too large.

Severity: Permanent negative completion.

Explanation: A command was issued which contained a pathname that specified a directory name which was longer than the maximum allowable 44 characters for an MVS data set name, or 1023 characters for an HFS file name.

System Action: The command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the command with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, mkd, rmd

---

501 Invalid file identifier in RNTO command.

Severity: Permanent negative completion.

Explanation: The RNTO command was issued to rename a data set or file. The data set (file) identifier is invalid or was not specified.

System Action: The data set or file is not renamed. FTP continues.

User or Operator Response: Re-issue the rename command observing the file naming conventions for MVS. For more information about MVS data set naming conventions, see the *OS/390 TCP/IP OpenEdition User's Guide*.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rnto

501 Invalid Jobid

Severity: Transient negative completion.

Explanation: A Command was issued requesting JES to delete a job. The jobid was not specified or it was incorrectly specified.

System Action: FTP continues.

User or Operator Response: Re-issue the command.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

501 Invalid user name '*' in USER command

Severity: Permanent negative completion.

Explanation: The client attempted to log on to the FTP server with a user ID of *. This is an incorrect user ID.

System Action: The user logon is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the USER command with the correct user ID.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: user

---

501 Member name too long. Use MVS naming conventions.

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with a data set name and member name requested, but the member name was longer than the MVS maximum length of 8 characters.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the command with the correct member name syntax.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 Mismatched quotes on directory name "*path name***"**

Severity: Permanent negative completion.

Explanation: A command was issued which started the pathname with a single quote while QUOTESOVERRIDE was TRUE. The FTP server expected a matching ending quote to be found at the end of the pathname, but no matching ending quote was found.

System Action: The command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the command with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, mkd, rmd

---

501 MKDIR fails: directory name too long

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new MVS PDS or a new HFS directory, but the requested directory name was longer than the HFS maximum length of 1023 for HFS directories, or longer than the MVS maximum length of 44 for MVS data sets.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Reissue the command with a valid directory name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

501 MKDIR fails: invalid directory name ''

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS, but the path name specified contained only quotation marks with no data set name.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Reissue the command with the data set name for the new PDS.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

501 No directory name entered

Severity: Permanent negative completion.

Explanation: The MKD command was entered to create a directory, or the RMD command was entered to delete a directory, but no directory was specified with the command.

System Action: The command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the MKD or RMD command, specifying a valid directory to be created or be deleted.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd, rmd

---

501 Nonexistent GDG data set *dsname*

Severity: Completion.

Explanation: The relative generation number for the Generation Data Group (GDG) data set is invalid. The number should be a negative integer, a positive integer (for a store command only), or a 0, enclosed in parentheses.

System Action: FTP continues.

User or Operator Response: Reissue the command with a valid relative number for the data set.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: rnto, retrieve, store_data

501 Partitioned data set may not be created within a partitioned data set

Severity: Permanent negative completion.

Explanation: The MKD command was entered to create a new PDS. The *path name* parameter was not in quotation marks, indicating that the path name should be appended to the current working directory; however, the current working directory was a partitioned data set, and a new partitioned data set cannot be created within the current working directory.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Either issue the CWD command to change the current working directory to a high-level qualifier rather than a PDS and then reissue the MKD command as before, or reissue the MKD command with the path name in quotation marks to prevent appending to the current working directory.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

501 Partitioned data set may not be deleted within a partitioned data set

Severity: Permanent negative completion.

Explanation: The RMD command was issued to delete a PDS. The path name specified with the RMD command was not in quotation marks, indicating that the path name should be appended to the current working directory to determine the name of the PDS to be deleted, but the current working directory was already a PDS and could not be appended to the path name.

System Action: The RMD command is rejected. Control returns to the client for further command processing.

User or Operator Response: Either issue a CWD command to change to current working directory and then issue the RMD command as before, or issue the RMD command with the fully qualified data set name in quotation marks to prevent the current working directory from being appended.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

501 password missing from PASS command

Severity: Permanent negative completion.

Explanation: The PASS command was issued to complete user log on, but no password was entered with the PASS command.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands, specifying a password with the PASS command.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

501 Patterns (% and *) may only be specified in the last qualifier when in directory mode

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered with a path name that contained the global file-name characters % and *. The global file-name characters were not in the last qualifier of the path name, and the MVS server was currently in directory mode. The global file-name characters can only be used in the last qualifier when the server is in directory mode.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Either correct the path name to have the global file-name characters in the last qualifier and reissue the command in directory mode, or issue the SITE DATASETMODE command to change the server to data set mode and then reissue the LIST or NLST command as is.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 Qualifier too long. Use MVS naming conventions.

Severity: Permanent negative completion.

Explanation: A LIST or NLST command was entered with a data set name that contained a qualifier that was longer than the MVS maximum of 8 characters.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Correct the data set name and reissue the command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

501 RMDIR fails: directory name too long

Severity: Permanent negative completion.

Explanation: The RMD command was issued to delete an MVS PDS or an HFS directory, but the requested directory name was longer than the HFS maximum length of 1023 for HFS directories, or longer than the MVS maximum length of 44 for MVS data sets.

System Action: The RMD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Reissue the command with a valid directory name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

501 RMDIR fails: invalid directory name ‘’

Severity: Permanent negative completion.

Explanation: The RMD command was issued to delete a PDS, but the path name contained only a set of quotation marks and no data set name.

System Action: The RMD command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the RMD command with the correct data set name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

501 Store unique not supported for GDG data sets.

Severity: Completion.

Explanation: The method of storing cannot be "store unique" when storing data in a Generation Data Group data set.

System Action: FTP continues.

User or Operator Response: Reissue the command after changing the storing method from store unique.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

501 Syntax error -- marker required for REST.

Severity: Completion.

Explanation: A restart (REST) command was received by the FTP server and there was no server marker at which the file transfer is to be restarted. The REST command must have a marker that is coordinated within the client program for a restart to occur.

Note: The REST command should not be issued directly by the user.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Determine why the client program did not send the marker.

Source Data Set: EZAFTPRF

Procedure Name: rest

---

501 Syntax error -- pathname required for DELE.

Severity: Completion.

Explanation: A delete command was received with no name specified.

System Action: FTP continues.

User or Operator Response: Reissue the command with a name to delete.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

---

501 Syntax error -- pathname required for RNFR.

Severity: Completion.

Explanation: A rename command was received with no name specified.

System Action: FTP continues.

User or Operator Response: Reissue the command with a name to rename.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: renamefr

---

501 too many bad passwords entered, PASS command disabled

Severity: Permanent negative completion.

Explanation: The PASS command was issued to complete log on to a user ID on the server system, but previous attempts to logon to this user ID with the incorrect password have caused the user ID to be locked out and log on for this user ID is no longer allowed.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: The FTP session must be terminated and reconnected to reset the number of bad password attempts for the user ID.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

501 TYPE command contains invalid parameter

Severity: Permanent negative completion.

Explanation: A TYPE B command was issued, but one or more of the parameters specifying DBCS options is invalid.

System Action: The TYPE command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the TYPE command with corrected parameters. Refer to IBM TCP/IP for MVS: User's Guide for information on the parameters for the TYPE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: dbcsOptions

---

501 TYPE command syntax error: format invalid

Severity: Permanent negative completion.

Explanation: The TYPE command was issued by the client. The TYPE command should have the syntax
TYPE *type {format opt1 opt2 opt3}* where each parameter is a single character. The FTP server determined that the value entered for the *format* parameter was not a single character. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the TYPE command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Re-enter the TYPE command with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: type_cmd

---

501 TYPE command syntax error: type invalid

Severity: Permanent negative completion.

Explanation: The TYPE command was issued by the client. The TYPE command should have the syntax
TYPE *type {format opt1 opt2 opt3}* where each parameter is a single character. The FTP server determined that the value entered for the *type* parameter was not a single character. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the TYPE command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Re-enter the TYPE command with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: type_cmd

---

501 TYPE command syntax error: 3rd parameter invalid

Severity: Permanent negative completion.

Explanation: The TYPE command was issued by the client. The TYPE command should have the syntax
TYPE *type {format opt1 opt2 opt3}* where each parameter is a single character. The FTP server determined that the value entered for the *opt1* parameter was not a single character. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the TYPE command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Re-enter the TYPE command with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: type_cmd

---

501 TYPE command syntax error: 4th parameter invalid

Severity: Permanent negative completion.

Explanation: The TYPE command was issued by the client. The TYPE command should have the syntax
TYPE *type {format opt1 opt2 opt3}* where each parameter is a single character. The FTP server determined that the value entered for the *opt2* parameter was not a single character. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the TYPE command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Re-enter the TYPE command with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: type_cmd

---

501 TYPE command syntax error: 5th parameter invalid

Severity: Permanent negative completion.

Explanation: The TYPE command was issued by the client. The TYPE command should have the syntax
TYPE *type {format opt1 opt2 opt3}* where each parameter is a single character. The FTP server determined that the value entered for the *opt3* parameter was not a single character. Refer to the *OS/390 TCP/IP OpenEdition User's Guide*, for information on the parameters of the TYPE command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: Re-enter the TYPE command with the correct syntax.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: type_cmd

---

501 Type remains Ascii NonPrint

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Ascii NonPrint.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected parameters. For information about valid TYPE parameters, IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains Big5

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Big5.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains Ebcdic NonPrint

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Ebcdic NonPrint.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains Hangeul

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Hangeul.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains Image

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Image (binary).

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with a corrected TYPe parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI EUC

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Extended Unix Code KANJI.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1978

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1978 shift-in ASCII

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978 shift-in ASCII.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1978 shift-in JISROMAN

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978 shift-in JISROMAN.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1983

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1983 shift-in ASCII

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983 shift-in ASCII.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI JIS 1983 shift-in JISROMAN

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983 shift-in JISROMAN.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE

parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KANJI Shift-JIS

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI Shift-JIS.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with a corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains KSC-5601

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Korean Standard Code KSC-5601.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

501 Type remains SChinese

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is SChinese.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

502 command *command* **not implemented**

Severity: Permanent negative completion.

Explanation: The client entered the FTP command *command*, but the FTP server does not support this command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

501 Type remains TChinese

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Traditional Chinese.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

501-Unknown mode *new_mode***.**
501 Data transfer mode remains *old_mode*

Severity: Permanent negative completion.

Explanation: The MODE command was entered to change the setting of MODE to *new_mode*; but *new_mode* was not a valid mode value.

System Action: The MODE command is rejected and control returns to the client for further command processing.

User or Operator Response: Reissue the MODE command with a valid value. Valid values are S (stream), B (block), or C (compressed).

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: mode

501-Unknown structure *new_stru*
501 Data structure remains *old_stru***.**

Severity: Permanent negative completion.

Explanation: The STRU command was entered to change the structure to *new_stru*, but *new_stru* was not a valid setting for the structure.

System Action: The STRU command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the STRU command with a valid value. Valid values are R (record structure) and F (file structure).

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: stru

501-unknown type *type_value*.

Severity: Permanent negative completion.

Explanation: The client entered the TYPE command to change the type setting to *type_value*, but the server did not recognize the *type_value* as a valid setting for TYPE.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Reissue the TYPE command with a valid TYPE value. For information about valid TYPE values, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

501 User name missing from USER command

Severity: Permanent negative completion.

Explanation: The client attempted to log on to the FTP server, but the name of the user ID to be logged in was missing from the USER command.

System Action: The user logon is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the USER command with the appropriate user ID.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: user

---

501 Wildcards not allowed in PDS name

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was issued requesting that members of a partitioned data set be listed. The name of the PDS contained one or more global file-name characters * or %. When listing members of a PDS, global file-name characters are allowed only in the member name, not in the PDS name.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Correct the data set name and reissue the request.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

502 command *command* not implemented

Severity: Permanent negative completion.

Explanation: The client entered the FTP command *command*, but the FTP server does not support this command.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: parse_cmd

---

502 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

502 error in host address ‘ *value*’

Severity: Permanent negative completion.

Explanation: The PORT command was issued by the client. The PORT command should have the syntax PORT *h1, h2, h3, h4, p1, p2* where *h1—h4* are the 4 integers that make up the host address and *p1* and *p2* are the 2 integers that make up the port address. When parsing the PORT command, the FTP server found a nonnumeric character in one of the 4 integers which make up the host address. *value* is the value of the parameter that the FTP server determined to be in error.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: If the command was issued by the user, reissue the command with the correct syntax. If the command was issued automatically by the FTP client program, contact the owner of the FTP client program for a possible programming error in the client program.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: onearg_cmd

---

502 error in port number ‘ *value*’

Severity: Permanent negative completion.

Explanation: The PORT command was issued by the client. The PORT command should have the syntax PORT *h1,h2,h3,h4,p1,p2* where *h1—h4* are the 4 integers that make up the host address and *p1* and *p2* are the 2 integers that make up the port address. When parsing the PORT command, the FTP server found a nonnumeric character in one of the 2 integers that make up the port address. *value* is the value of the parameter that the FTP server determined to be in error.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: If the command was issued by the user, reissue the command with the correct syntax. If the command was issued automatically by the FTP client program, contact the owner of the FTP client program for a possible programming error in the client program.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: onearg_cmd

502 PORT command syntax error - insufficient digits in host-port

Severity: Permanent negative completion.

Explanation: The PORT command was issued by the client. The PORT command should have the syntax PORT *h1,h2,h3,h4,p1,p2* where *h1—h4* are the 4 integers that make up the host address and *p1* and *p2* are the 2 integers that make up the port address. When parsing the PORT command, the FTP server did not find a total of 6 integer values to make up the host address and port address.

System Action: The command is rejected. Control is returned to the client for further command processing.

User or Operator Response: If the command was issued by the user, reissue the command with the correct syntax. If the command was issued automatically by the FTP client program, contact the owner of the FTP client program for a possible programming error in the client program.

System Programmer Response: None.

Source Data Set: EZAFTPPA

Procedure Name: onearg_cmd

---

503 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

503 PASS command must be preceded by USER

Severity: Permanent negative completion.

Explanation: The PASS command was entered to complete log on of a user ID on the server system, but the USER command has not been issued to specify which user ID to logon to. The USER command must be issued to specify the user ID before the PASS command can be issued.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Issue the USER command to specify the user ID of the server system to logon to, then reissue the PASS command with the password for this user ID.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

504 Block mode not implemented for type B.

Severity: Completion.

Explanation: The data type B (double Byte) is not supported while in block transmission mode.

System Action: FTP continues.

User or Operator Response: Either change to mode s (stream) or change the data type to EBCDIC and reissue the command.

System Programmer Response:

Source Data Set: EZAFTPMJ, EZAFTPRR

Procedure Name: jesGet, jesPut, retrieve

---

504 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

504 Compress mode not implemented for type B.

Severity: Completion.

Explanation: The data type B (double Byte) is not supported while in compress transmission mode.

System Action: FTP continues.

User or Operator Response: Either change to mode s (stream) or change the data type to EBCDIC and reissue the command.

Source Data Set: EZAFTPMJ, EZAFTPRR

Procedure Name: jesGet, jesPut, retrieve

---

504 Error - existing kanji type invalid

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is for a DBCS data type, but an internal error has occurred and the current data type is no longer valid.

System Action: The FTP connection with the client is terminated.

User or Operator Response: Reconnect to the FTP server and issue the desired TYPE command. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide. If this reply is received repeatedly, make a note of the last few FTP commands you have issued that cause this result, and contact your system programmer

System Programmer Response: Contact the IBM Support Center with output from the FTP server trace, if available.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

504 Error - existing type invalid

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. FTP tried to continue with the current data transfer type, but an internal error has occurred and the current data type is no longer valid.

System Action: The FTP connection with the client is terminated.

User or Operator Response: Reconnect to the FTP server and issue the desired TYPE command. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide. If this reply is received repeatedly, make a note of the last few FTP commands you have issued that cause this result, and contact your system programmer

System Programmer Response: Contact the IBM Support Center with output from the FTP server trace, if available.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Only local byte size allowed is 8

Severity: Permanent negative completion.

Explanation: The client entered the TYPE command with a parameter of “L byte_size” to change the type setting to the specified local byte size. The byte_size specified by the command was a value other than 8, but the only local byte size supported by the server is 8.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Reissue the TYPE command with a valid local byte size.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-Page structure not implemented
504 Data structure remains ***old_stru*****.**

Severity: Permanent negative completion.

Explanation: The STRU command was entered to change the structure to PAGE, but PAGE structure is not supported by the FTP server.

System Action: The STRU command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the STRU command with a valid value. Valid values are R (record structure) and F (file structure).

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: stru

---

504 Restart not allowed for store unique.

Severity: Completion.

Explanation: A restart (REST) command was received by the FTP server and the storage method is "store unique." This combination is not allowed for a restarted store operation.

System Action: FTP continues.

User or Operator Response: Change from store unique method and reissue the restart command.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_unique

---

504 Restart requires Block or Compressed transfer mode.

Severity: Permanent negative completion.

Explanation: You attempted to restart a checkpointed data transfer command and the transfer mode is not block or compressed.

System Action: The data transfer command is not restarted. FTP continues.

User or Operator Response: Change the transfer mode to block or compressed and re-enter the restart command.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rest

---

504 Restart requires EBCDIC data type.

Severity: Permanent negative completion.

Explanation: You attempted to restart a checkpointed data transfer command and the data type is not EBCDIC.

System Action: The data transfer command is not restarted. FTP continues.

User or Operator Response: Change the data type to EBCDIC and re-enter the restart command.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rest

---

504 Restart requires filetype=SEQ.

Severity: Permanent negative completion.

Explanation: You attempted to restart a checkpointed data transfer command and the filetype is not SEQ.

System Action: The data transfer command is not restarted. FTP continues.

User or Operator Response: Change the filetype to SEQ and re-enter the restart command.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rest

---

504 STAT file-identifier: not implemented

Severity: Completion.

Explanation: The STAT subcommand was entered with the *file-identifier* parameter. The *file-identifier* parameter of the STAT subcommand is not supported by the FTP server.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: ftpstat

504 Struct R implemented with stream mode only.

Severity: Permanent negative completion.

Explanation: You attempted to get or store a data set while the data structure is defined as record and the transfer mode is not stream mode.

System Action: The request is rejected. FTP continues.

User or Operator Response: Change the transfer mode to stream (MODE S) or change the data structure to file (STRU F).

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRR, EZAFTPRS

Procedure Name: jesGet, jesPut, retrieve, store_data

504-TYPE has unknown format ***format*****.**

Severity: Permanent negative completion.

Explanation: The client entered the TYPE command in the format "TYPE type_value *format*" to change the type setting, but the *format* specified was not recognized by the FTP server.

System Action: The TYPE command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the TYPE command with a valid format value. For information about valid parameters for the TYPE command, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-TYPE has unsupported format *format*

Severity: Permanent negative completion.

Explanation: A TYPE A or TYPE E command was received by the FTP server, but format *format* is not supported. The only format parameter supported is N (Non-PrinT), which is also the default.

System Action: The TYPE command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the corrected TYPE command.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-TYPE not Supported. Translation table not Loaded.

Severity: Permanent negative completion.

Explanation: A TYPE B command was received for a DBCS language, but the the required DBCS translation table is not available to the FTP server.

System Action: The TYPE command is rejected. Control returns to the client for further command processing.

User or Operator Response: Contact the system programmer at the server system to get the necessary DBCS tables loaded.

System Programmer Response: Ensure that the TCP/IP.DATA file contains a LOADDBCSTABLES statement that correctly specifies the DBCS languages whose translation tables are to be loaded. For information about the LOADDBCSTABLES parameters, see *OS/390 TCP/IP OpenEdition Configuration Guide*. If the TCPIP.DATA file is changed, the FTP server must be restarted to recognize the change.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-Type not supported. Unable to load Japanese translation tables.

Severity: Permanent negative completion.

Explanation: The FTP server received a TYPE B command but an error occurred when the server tried to load the required translation tables for this client.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Contact the system programmer for the FTP server's system.

System Programmer Response: Look in the system log for the FTP server for message EZY2721E or EZY2722E. Probable causes of the problem include unable to open the TCPKJBIN file containing the translation tables, insufficient storage available for the translate tables, or the file format is invalid.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-Type not supported. Unable to load Korean translation tables.

Severity: Permanent negative completion.

Explanation: The FTP server received a TYPE B command but an error occurred when the server tried to load the required translation tables for this client.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Contact the system programmer for the FTP server's system.

System Programmer Response: Look in the system log for the FTP server for message EZY2723E or EZY2724E. Probable causes of the problem include unable to open the TCPHGBIN file containing the translation tables, insufficient storage available for the translate tables, or the file format is invalid.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-Type not supported. Unable to load simplified Chinese translation tables.

Severity: Permanent negative completion.

Explanation: The FTP server received a TYPE B command but an error occurred when the server tried to load the required translation tables for this client.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Contact the system programmer for the FTP server's system.

System Programmer Response: Look in the system log for the FTP server for message EZY2726E or EZY2727E. Probable causes of the problem include unable to open the TCPSCBIN file containing the translation tables, insufficient storage available for the translate tables, or the file format is invalid.

Source Data Set: EZAFTPRT

Procedure Name: xtype

504-Type not supported. Unable to load traditional Chinese translation tables.

Severity: Permanent negative completion.

Explanation: The FTP server received a TYPE B command but an error occurred when the server tried to load the required translation tables for this client.

System Action: The TYPE command is rejected. Control returns to the client for further commands.

User or Operator Response: Contact the system programmer for the FTP server's system.

System Programmer Response: Look in the system log for the FTP server for message EZY2726E or EZY2727E. Probable causes of the problem include unable to open the TCPCHBIN file containing the translation tables, insufficient storage available for the translate tables, or the file format is invalid.

Source Data Set: EZAFTPRT

Procedure Name: xtype

---

504 Type remains Ascii NonPrint

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Ascii NonPrint.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains Big5

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Big5.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains Ebcdic NonPrint

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Ebcdic NonPrint.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains Hangeul

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Hangeul.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains Image

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Image (binary).

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with a corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI EUC

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Extended Unix Code KANJI.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1978

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE

parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1978 shift-in ASCII

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978 shift-in ASCII.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1978 shift-in JISROMAN

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1978 shift-in JISROMAN.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1983

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1983 shift-in ASCII

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983 shift-in ASCII.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI JIS 1983 shift-in JISROMAN

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI JIS 1983 shift-in JISROMAN.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KANJI Shift-JIS

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is KANJI Shift-JIS.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with a corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

504 Type remains KSC-5601

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Korean Standard Code KSC-5601.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

504 Type remains SChinese

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is SChinese.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

521 *"new_directory"* **data set already exists**

Severity: Permanent negative completion.

Explanation: The MKD command was entered to create a new PDS with the name *new_directory*, but a data set with that name already exists at the server system.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Change the name of the *new_directory* to a unique name and reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

504 Type remains TChinese

Severity: Permanent negative completion.

Explanation: The FTP server received an invalid TYPE command. The current data transfer type is Traditional Chinese.

System Action: The FTP transfer data type is unchanged. Control returns to the client for further commands.

User or Operator Response: If you want to change the data transfer type, reissue the TYPE command with corrected TYPE parameters. For information about valid TYPE parameters, see IBM TCP/IP for MVS: User's Guide.

System Programmer Response: None.

Source Data Set: EZAFTPRT

Procedure Name: report_old_type

---

521 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

521 *"new_directory"* **data set already exists**

Severity: Permanent negative completion.

Explanation: The MKD command was entered to create a new PDS with the name *new_directory*, but a data set with that name already exists at the server system.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Change the name of the *new_directory* to a unique name and reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

530 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

530 Excessive bad passwords entered - USER command disabled

Severity: Permanent negative completion.

Explanation: The client attempted to log on to the FTP server system, but the user ID being used by the client had too many logon attempts that were specified with an incorrect password and this user ID has been disabled.

System Action: The user logon is rejected. Control returns to the client for further command processing.

User or Operator Response: The FTP session must be terminated and reconnected to reset the number of bad password attempts.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: user

---

530 Logon attempt by ' *user_id*" rejected by user exit.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the user ID *user_id*, but the user exit FTCHKPWD loaded by the server rejected the log on attempt of this user ID.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system.

System Programmer Response: If *user_id* is to be allowed to logon to the server, the FTCHKPWD user exit must be modified to allow user log on.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

530 Logon attempt by ' *user_id*' rejected.

Severity: Completion.

Explanation: The FTP server could not successfully validate the user's user_Id and password. The user logon is rejected.

System Action: The user logon is rejected. FTP continues.

User or Operator Response: This reply is preceded by a reply which indicates the reason for the failure. Correct the error indicated by the preceding reply. If necessary, contact the system programmer for the appropriate security authorization to access the FTP server.

System Programmer Response: If necessary, authorize the user to access the FTP server.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

530 new password format invalid

Severity: Permanent negative completion.

Explanation: The PASS command was issued using the format old_password/new_password/new_password to change the password of the user ID, but the second "/" could not be found in the password entered with the PASS command.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands, using the correct format of the "old_password/new_password/new_password" on the PASS command.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

530 new passwords are not the same

Severity: Permanent negative completion.

Explanation: The PASS command was issued using the format old_password/new_password/new_password to change the password of the user ID, but the second "new password" was not identical to the first "new password". Both "new passwords" must be the same.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands, using the same password for both occurrences of "new password."

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

530 Not logged in.

Severity: Permanent negative completion.

Explanation: A command was issued to request a service from the FTP server, but the client was not currently logged on to a valid user ID for the server system. The command entered required the user to be logged on to a valid user ID.

System Action: The command is rejected. Control returns to the client for further command processing.

User or Operator Response: Log on to a valid user ID for the server system using the USER and PASS commands, then reissue the command. See the *OS/390 TCP/IP OpenEdition User's Guide* for more information.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: jesDelete, jesGet, jesPut, delete, renamefr, rest, rnto, retrieve, store_data

---

530-RACF authorization failed: rc *saf_rc/racf_rc/racf_re***.**

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the server system with user ID *user_id*. The RACF security system rejected the user log on, but the FTP server was unable to interpret the reason for the log on rejection.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system.

System Programmer Response: Determine the reason for the RACF rejection. The return codes in the 530 response are the SAF return code (*saf_rc*), the RACF return code (*racf_rc*), and the RACF reason code (*racf_re*).

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The new password is invalid.

Severity: Permanent negative completion.

Explanation: The PASS command was issued using the format old_password/new_password/new_password to change the password of the user ID, but the "new password" was not acceptable to the server system.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands using a valid "new password." If necessary, contact the system programmer of the server system for information on valid passwords for the server system.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The password has expired.

Severity: Permanent negative completion.

Explanation: The PASS command was issued to logon to user ID *user_id*, but the password for user ID *user_id* had expired.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: The password for *user_id* at the server system must be changed. This can be done in one of the following ways:

- Reissue the FTP USER and PASS commands using the format "old_password/new_password/new_password" to change the password during FTP Logon.
- If you have access to the server system, logon to the user ID and change the password.
- Contact the system programmer at the server system to get the password updated.

Once the password for *user_id* has been changed, reissue the USER and PASS commands with the new password.

System Programmer Response: If necessary, update the password for user ID *user_id*.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The password is not authorized.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to user ID *user_id*, but the password used for Logon was not the correct password for that user ID.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands with the correct password.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The port of entry may not be used this day, or at this time of day.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the server system user ID *user_id*, but the logon was rejected because the RACF configuration at the server system restricted the usage of the port of entry the client was using.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system for information on the RACF restrictions of the server system. If possible, retry the FTP logon at a time when this port of entry is allowed.

System Programmer Response: If necessary, update the RACF definitions to allow this port of entry access.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user is not authorized to access the system on this day, or at this time of day.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the server system with user ID *user_id*, but the RACF system at the server denied access to user ID *user_id* at this time.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer at the server system for information on when this user ID is allowed access to the system. Retry the Logon when the user ID is allowed access to the system.

System Programmer Response: If necessary, update the RACF definitions to allow *user_id* access to the server system.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user is not authorized to the port of entry.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the FTP server system with user ID *user_id*, but RACF rejected *user_id* because it was not authorized for the port of entry that was being accessed.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system for authorization.

System Programmer Response: If necessary, update the RACF definitions to allow *user_id* access to the port of entry.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user is not authorized to use this application.

Severity: Permanent negative completion.

Explanation: The client was attempting to logon to the FTP server using user ID *user_id*, but RACF rejected the logon attempt because *user_id* was not authorized to access the FTP server application.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system.

System Programmer Response: If necessary, update the RACF definitions to allow *user_id* access to the FTP server application.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user is not defined to the group.

Severity: Permanent negative completion.

Explanation: The client was attempting to logon to the FTP server using user ID *user_id*, but RACF rejected the logon because *user_id* was not defined to the necessary RACF group for accessing the FTP server.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system.

System Programmer Response: If necessary, update the RACF definitions to add *user_id* to the necessary group to access the FTP server.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user is not defined to the system.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the FTP server using user ID *user_id*, however *user_id* was not a valid user ID for the server system.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Reissue the USER and PASS commands with a valid user ID.

System Programmer Response: None.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user's access has been revoked.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the FTP server using user ID *user_id*; however, RACF rejected the logon attempt because *user_id*'s access to the server system had been revoked.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer for the server system to allow *user_id* to logon to the server system.

System Programmer Response: If necessary, reinstate *user_id*'s access to the system.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-The user's access to the specified group has been revoked.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the FTP server using user ID *user_id*; however, RACF rejected the logon because *user_id*'s access to the necessary RACF group had been revoked.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer for the server system to get the RACF access for *user_id* reinstated.

System Programmer Response: If necessary, reinstate *user_id*'s access to the necessary group.

Source Data Set: EZAFTPRA

Procedure Name: pass

530-Verify failed by the installation exit routine.

Severity: Permanent negative completion.

Explanation: The client attempted to logon to the server system using user ID *user_id*; however, one of the RACF installation exit routines rejected the logon attempt.

System Action: Logon to the user ID on the server system is rejected. Control is returned to the client for further command processing.

User or Operator Response: Contact the system programmer of the server system for the necessary RACF authority to logon to the server system.

System Programmer Response: If necessary, change the RACF installation exit routine to allow access for *user_id*.

Source Data Set: EZAFTPRA

Procedure Name: pass

---

530 You must first login with USER and PASS

Severity: Permanent negative completion.

Explanation: A command was issued to request a service from the FTP server, but the client was not currently logged on to a valid user ID for the server system. The command entered required the user to be logged on to a valid user ID.

System Action: The command is rejected. Control returns to the client for further command processing.

User or Operator Response: Log on to a valid user ID for the server system using the USER and PASS commands, then reissue the command. See the *OS/390 TCP/IP OpenEdition User's Guide* for more information.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, list2, mkd, pwd, rmd

---

550 *cmd* **fails:** *dsname***. User not authorized.**

Severity: Permanent negative completion.

Explanation: A command was issued to store a data set. (*cmd* is STOR, STOU, or APPE.) The requested data set is protected by a security system such as RACF, and the user is not authorized to write to the data set.

System Action: The data set is not stored. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to store into the data set.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

550 *dsname* **is a partitioned data set and no member was specified on the** *cmd* **command.**

Severity: Permanent negative completion.

Explanation: The server attempted to store to a partitioned data set (PDS) but did not provide a member name. *cmd* is STOR, STOU, or APPE.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Provide a member name for the PDS.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 *dsname* **is a physical sequential data set and a member was specified on the command.**

Severity: Permanent negative completion.

Explanation: You requested a member of a data set but the data set that was named is a physical sequential data set (PDS) and does not have members.

System Action: The data set is not stored. FTP continues.

User or Operator Response: Provide the correct name of the PDS that contains the member requested.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: file_check, find_file, find_unique_file

---

550 *dsname* **is not on a direct access volume. It may not be renamed.**

Severity: Permanent negative completion.

Explanation: The data set to be renamed is not on a direct access volume.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: Make sure that the data set name is for a data set that can be renamed. A tape data set is an example of data set that cannot be renamed.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: renamefr

---

550 " *directory***" data set does not exist.**

Severity: Permanent negative completion.

Explanation: The RMD was issued to delete the partitioned data set *directory*, but the server could not find the data set to delete it.

System Action: The RMD command is rejected. Control returns to the client for further command processing.

User or Operator Response: Verify that the correct data set name was entered. If necessary, reissue the command with the correct data set name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

550 Allocation for remote destination failed.

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a SYSOUT data set for the purpose to sending a data set to a remote destination (see SITE DEST). The allocation was unsuccessful.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *seq_alloc_dest*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

550 Allocation of *dsname* **failed while executing RETR command.**

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a data set on a DASD volume. The dynamic allocation was unsuccessful.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *alloc_dasd*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

550 Allocation of *dsname* **failed while executing** *cmd* **command.**

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a new data set to store data. The dynamic allocation was unsuccessful. *cmd* is STOR, STOU, or APPE.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *seq_create_file*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Cannot create unique data set name for store unique of *dsname***.**

Severity: Permanent negative completion.

Explanation: A unique name cannot be created to store a data set using the store unique command. The server appends numbers 1 to 999 to the end of the data set name to create a new name. This set of names is exhausted.

System Action: The command is rejected. FTP continues.

User or Operator Response: Change the file name to allow the server a set of names to search for uniqueness.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: find_unique_file

550 Cannot create unique file name for store unique of *filename.***.**

Severity: Permanent negative completion.

Explanation: A unique name cannot be created to store a file in the hierarchical file system using the store unique command. The server appends numbers 1 to 999 to the end of the file name to create a new name. This set of names is exhausted.

System Action: The command is rejected. FTP continues.

User or Operator Response: Change the file name to allow the server a set of names to search for uniqueness.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: find_uhfs_file

---

550 Cannot create unique member name for append to *dsname*

Severity: Permanent negative completion.

Explanation: An append was requested and the data set is a member of a partitioned data set (PDS). Before the append can occur, the member must be copied forward in the PDS and in the process is temporarily given a unique name. This process was attempted 999 times and a unique name was not found.

System Action: The append is not performed.

User or Operator Response: The unique names that are used for the copy are created by appending 1, then 2, and so forth to the end of member name (called the base name). If there are members of the PDS that have the same base name as the one for the append and the members are no longer needed, then delete them and re-issue the append request.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: find_unique_file

---

550 Cannot create unique member name for store unique of *dsname***.**

Severity: Permanent negative completion.

Explanation: A unique name cannot be created to store a member of a partitioned data set (PDS) using the store unique command. The server appends numbers 1 to 999 to the end of the member name to create a new name. This set of names is exhausted.

System Action: The command is rejected. FTP continues.

User or Operator Response: Change the member name to allow the server a set of names to search for uniqueness.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: find_unique_file

---

550 Cannot rename from partitioned data set *dsname* **to a different partitioned data set** *dsname***.**

Severity: Permanent negative completion.

Explanation: You have attempted to rename a member of a partitioned data set (PDS). The new name is a member of a different PDS. A member of a PDS can be renamed only to a new member name within the same PDS.

System Action: The rename is not performed.

User or Operator Response: Provide a new name that is a name in the same PDS.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rnto

---

550 *cmd* **cmd failed :** *error*

Severity: Permanent negative completion.

Explanation: The FTP client issued the command *cmd*. While processing the command, the FTP server issued a C runtime library function which did not complete successfully. *error* is the error message returned by the C runtime library.

System Action: The command *cmd* is not executed.

User or Operator Response: Re-issue the command. If the problem persists, contact the system programmer with the error message.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, list2

---

550 *cmd* **cmd failed. No files found.**

Severity: Permanent negative completion.

Explanation: The LIST or NLST command issued for an HFS directory failed because the directory did not contain any files.

System Action: FTP continues processing with the next command

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 *cmd* **command failed - pipe error:** *error*

Severity: Permanent negative completion.

Explanation: The client issued the LIST or NLST command to list files in an HFS directory. While processing the *cmd* command, the server issued the C runtime library routine pipe(), which did not complete successfully. *error* is the error message returned by the C runtime library.

System Action: The *cmd* command is rejected. The server waits for the next command to be entered by the client.

User or Operator Response: Reissue the command. If the problem persists, contact the System Programmer.

System Programmer Response: Correct the problem indicated by *error*.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 *cmd* **command failed - popen error:** *error*

Severity: Permanent negative completion.

Explanation: The client issued the LIST or NLST command to list files in an HFS directory. While processing the *cmd* command, the server issued the C runtime library routine popen(), which did not complete successfully. *error* is the error message returned by the C runtime library.

System Action: The *cmd* command is rejected. The server waits for the next command to be entered by the client.

User or Operator Response: Reissue the command. If the problem persists, contact the System Programmer.

System Programmer Response: Correct the problem indicated by *error*.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 *cmd* **failed: Unknown HOME directory**

Severity: Permanent negative completion.

Explanation: A command was issued that required the server to resolve the user's HOME directory name (for example, a pathname was entered that began with the directory notation ~/) but the server was unable to determine the user's HOME directory.

System Action: The command is rejected. FTP continues.

User or Operator Response: Verify that the OMVS user_id has a HOME directory. If necessary, define a HOME directory for the user_id.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, mkd, rmd

---

550 *cmd* **fails:** *filename* **does not exist.**

Severity: Permanent negative completion.

Explanation: The file that is named does not exist. The retrieve, delete, and rename subcommands require that the file exists.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying a name of a file exists.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR

Procedure Name: hfs_delete, hfs_renamefr, retrieve

---

550 *cmd* **fails:** *filename* **is a character special file.**

Severity: Permanent negative completion.

Explanation: The file that is named is a character special file. The file transfer, delete, and rename subcommands do not support files of this type.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying a name of a file that is not a character special file.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: hfs_delete, hfs_renamefr, retrieve, find_hfs_file

---

550 *cmd* **fails:** *filename* **is an unknown type.**

Severity: Permanent negative completion.

Explanation: The file that is named is an unknown type. The file transfer, delete, and rename subcommands do not support files of this type.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying a name of a file that is not an unknown file type.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: hfs_delete, hfs_renamefr, retrieve, find_hfs_file

---

550 *cmd* **fails:** *filename* **is a directory and is not empty.**

Severity: Permanent negative completion.

Explanation: The file that is named is a directory. The delete and rename subcommands support a directory name but the directory must be empty.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying an empty directory.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: hfs_delete, hfs_rnto

---

550 *cmd* **fails:** *filename* **is a directory.**

Severity: Permanent negative completion.

Explanation: The file that is named is a directory. The RETR, STOR, STOU, and APPE commands do not support file transfer for a directory file.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying a name of a file that is not a directory.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, find_hfs_file

---

550 *cmd* **fails:** *filename* **is a pipe or FIFO.**

Severity: Permanent negative completion.

Explanation: The file that is named is a pipe or FIFO special file. The file transfer, delete, and rename subcommands do not support files of this type.

System Action: The command is rejected. FTP continues.

User or Operator Response: Re-issue the command specifying a name of a file that is not a pipe or FIFO special file.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: hfs_delete, hfs_renamefr, retrieve, find_hfs_file

---

550 *cmd* **fails:** *pathname* **is a directory.**

Severity: Completion.

Explanation: An attempt was made to rename an HFS file to a name that is an existing directory.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: hfs_rnto

550 *cmd* **fails:** *pathname* **is not a directory.**

Severity: Completion.

Explanation: An attempt was made to rename an HFS directory to a name that is an existing regular file.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: hfs_rnto

---

550 *command* **fails:** *path name*. **User not authorized**

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was issued to list members of a partitioned data set, but the requested data set was protected by a security system such as RACF, and the user was not authorized to read the data set.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Contact the owner of the data set for authorization to read the data set.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

550 Data set *dsname* **is migrated and NoAutoRecall is specified.**

Severity: Permanent negative completion.

Explanation: A command was issued to process a data set, but the data set is migrated and the FTP server is currently in NoAutoRecall mode.

System Action: The command is rejected.

User or Operator Response: Issue the "SITE AUTORECALL" command to allow the data set to be recalled, and then re-issue the FTP command.

System Programmer Response: None.

Source Data Set: EZAFTPRD, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: cwd2, list2, delete, renamefr, file_check, find_file, find_unique_file

---

550 Data set *dsname* **not found**

Severity: Permanent negative completion.

Explanation: The server attempted to retrieve a data set (that is, a physical sequential data set or member of a partitioned data set). The data set was not found.

System Action: The command is rejected. FTP continues.

User or Operator Response: Ensure that the data set exists and that the fully qualified name used for the retrieve is correct.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

550 *dsname* **is not on a direct access volume. It may not be deleted.**

Severity: Permanent negative completion.

Explanation: The data set to be deleted is not on a direct access volume.

System Action: The data set is not deleted. FTP continues.

User or Operator Response: Make sure that the data set name is for a data set that can be deleted. A tape data set is an example of data set that cannot be deleted.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

---

550 *dsname* **used exclusively by someone else.**

Severity: Permanent negative completion.

Explanation: Either a data transfer (retrieve or store), a delete, or a rename of a member of a partitioned data set (PDS) was requested and the member requested is currently in use.

System Action: The operation is not performed.

User or Operator Response: Try the operation later when no one else is using the member of the PDS.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: delete, rnto, retrieve, store_data

---

550 DELE fails: *dsname* **does not exist.**

Severity: Permanent negative completion.

Explanation: The DELE command was issued to delete a data set. The delete was unsuccessful because the data set does not exist.

System Action: None. FTP continues.

User or Operator Response: Make sure that the correct data set name is entered and that the data set is catalogued.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

550 DELE fails: *dsname*. **User not authorized.**

Severity: Permanent negative completion.

Explanation: The DELE command was issued to delete a data set. The requested data set is protected by a security system such as RACF, and the user is not authorized to delete the data set.

System Action: The data set is not deleted. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to delete it.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

---

550 directory remains "dsname".

Severity: Permanent negative completion.

Explanation: A CWD command was received to change the current working directory, but the server was not able to successfully change the current working directory. This reply is preceded by other replies with additional information about the error.

System Action: The working directory is not changed. FTP continues.

User or Operator Response: Correct the errors indicated by the replies which preceded this one.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2

---

550 error allocating *new_directory*

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create the new PDS *new_directory*, but the server was unable to allocate the PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Contact the system programmer for the server system.

System Programmer Response: Determine why the PDS could not be allocated and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 Error allocating storage for gtlist

Severity: Permanent negative completion.

Explanation: The LIST or NLST command was entered to list MVS files. The server was unable to get the necessary storage for the catalog list.

System Action: Processing of the LIST / NLST command is terminated. The FTP server continues processing with the next command.

User or Operator Response: Retry the list. If the problem persists, contact the system programmer.

System Programmer Response: Allocate additional storage for the address space.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 Error allocating storage for list

Severity: Permanent negative completion.

Explanation: The FTP server was not able to allocate the storage necessary to process the LIST or NLST command.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the FTP server system.

System Programmer Response: The FTP server might need to be started with a larger region size.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 Error allocating tape data set *dsname*

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a data set on a tape volume. The dynamic allocation was unsuccessful.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *alloc_tape*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

550 Error deleting migrated data set *dsname*

Severity: Permanent negative completion.

Explanation: The services of DFHSM were invoked to delete a data set that is migrated. DFHSM cannot delete the data set.

System Action: The data set is not deleted. FTP continues.

User or Operator Response: Recall the data set and retry the command.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: delete

---

550 Error mounting volume containing data set *dsname*

Severity: Permanent negative completion.

Explanation: The volume containing the requested data set was not mounted to the FTP server system. The FTP server attempted to have the volume mounted, but was unable to get the volume mounted.

System Action: The command is rejected.

User or Operator Response: Re-issue the command. If the problem persists, contact the system programmer for the FTP server system.

System Programmer Response: Determine why the requested volume cannot be mounted to the MVS system.

Source Data Set: EZAFTPRD, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: list2, delete, renamefr, file_check, find_file, find_unique_file

550 Error opening *data_set_name*

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested members of a PDS, but the FTP server was unable to open the PDS to read the directory. This can be a temporary problem, for example the data set is currently in use, or it can be a permanent error with the data set.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Correct the problem with the data set.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 error processing PASS command : *error*

Severity: Permanent negative completion.

Explanation: The client has entered the USER and PASS commands to logon to the FTP server. The FTP server encountered an error attempting to setup the OMVS environment for the requested user_id. *error* is the error message returned by the C runtime library.

System Action: Logon to the FTP server is rejected.

User or Operator Response: Contact the system programmer

System Programmer Response: Correct the error indicated by *error*

Source Data Set: EZAFTPRA

Procedure Name: pass

---

550 error reading PDS directory

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested members of a partitioned data set, but the FTP server was unable to read the directory of the PDS.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Correct the problem with the data set.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 Error recalling data set *data_set*

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested members of a partitioned data set. The PDS was migrated and needed to be recalled to read the PDS directory. The FTP server attempted to recall the data set, but was unable to successfully recall the data set.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the data set cannot be recalled and correct the problem.

Source Data Set: EZAFTPRD, EZAFTPRF, EZAFTPRS

Procedure Name: list2, delete, renamefr, find_file, find_unique_file

---

550 Error recalling data set *data_set* **(rc=***rc***)**

Severity: Permanent negative completion.

Explanation: A RETR request was received for a data set. The data set was migrated and needed to be recalled. The FTP server attempted to recall the data set, but was unable to successfully recall the data set.

System Action: The RETR command is rejected.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: Determine why the data set cannot be recalled and correct the problem.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

550 File *filename* **not found.**

Severity: Permanent negative completion.

Explanation: The server attempted to retrieve an hierarchical file system (HFS) file. The file was not found.

System Action: The command is rejected. FTP continues.

User or Operator Response: Ensure that the file exists and that you have access to all of the directories in the path to the file.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

550 *file_name* **request nonexistent member to be sent.**

Severity: Permanent negative completion.

Explanation: The member that was named is not a member of the partitioned data set (PDS) and cannot be retrieved.

System Action: The data set is not sent. FTP continues.

User or Operator Response: Reissue the command with the corrected PDS or member name.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

550 *file_name* **requests a non-existent partitioned data set. Use MKD command to create it.**

Severity: Permanent negative completion.

Explanation: The store request names a partitioned data set (PDS) that does not exist.

System Action: The data set is not sent. FTP continues.

User or Operator Response: Either name an exiting PDS or use MKD command to create a PDS into which the member will be stored.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: find_file, find_unique_file

550 Index *number* is greater than number of spool files for *&jobid.*

Severity: Permanent negative completion.

Explanation: User attempted to retrieve a specific spool file from JES using the FTP GET command. JES output files are numbered, and the DATASET_NUMBER specified for the spool file desired was not found for this job.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Re-issue the FTP GET command with the correct JOBID and DATASET_NUMBER.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 Invalid cancel request

Severity: Permanent negative completion.

Explanation: A DELETE command was issued. The command was not formed correctly.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Invalid jobname/Jobid combination

Severity: Permanent negative completion.

Explanation: JES could not perform the requested function because the exact JOBNAME and JOBID combination specified in the command could not be located.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 JES cannot find output for *&jobid.*

Severity: Permanent negative completion.

Explanation: User attempted to retrieve data for a specific jobid. The jobid specified in the command was not found.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 JES cannot find output for *&jobid.*, JesPutGet aborted

Severity: Permanent negative completion.

Explanation: User attempted to retrieve data for a specific jobid. The jobid specified in the command was not found.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 JES internal reader allocation failed

Severity: Permanent negative completion.

Explanation: In preparation of submitting a job to MVS the Internal Reader needed to be allocated. The allocation of the Internal Reader failed.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

550 JES internal reader open failed

Severity: Permanent negative completion.

Explanation: In preparation of submitting a job to MVS the Internal Reader needed to be opened. The request to open the Internal Reader failed.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

550 JES is unable to provide spool data set name now

Severity: Permanent negative completion.

Explanation: User attempted to retrieve spool files from JES. Requested dataset is open.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Issue the command later

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

550 JES is unable to provide spool data set name now, JesPutGet aborted

Severity: Permanent negative completion.

Explanation: User attempted to retrieve spool files from JES. Requested dataset is open.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Issue the command later

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 JES spool file allocation failed for *dsname*

Severity: Permanent negative completion.

Explanation: An allocate of a spool file for get processing failed.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesRetrieveSpoolFile

---

550 JES spool file open failed for *dsname*

Severity: Permanent negative completion.

Explanation: While attempting to open a spool file for processing, an error occurred.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesRetrieveSpoolFile

---

550 JESPUT failed, could not allocate receive buffer

Severity: Permanent negative completion.

Explanation: A buffer was not available while attempting to retrieve a spool output.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

550 JesPutGet aborted

Severity: Permanent negative completion.

Explanation: Refer to replies prior to this 550 reply.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jes_submit_job

---

550 JesPutGet aborted, internal error

Severity: Permanent negative completion.

Explanation: A GET command was issued with a SITE FILETYPE=JES argument in effect. The GET command requested the output to be automatically retrieved. The output was not retrieved because of an internal processing error in FTP.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Attempt to retrieve the job from JES explicitly. If this problem prevails, contact your system support personnel for assistance.

System Programmer Response: None: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

550 JesPutGet aborted, job not found

Severity: Permanent negative completion.

Explanation: A GET command was issued with a SITE FILETYPE=JES argument in effect. The GET command requested the output to be automatically retrieved. The output was not available to be retrieved within the time period allowed for the JOB to complete execution.

System Action: FTP server continues normal execution. The FTP client waits for the next command input. The job submitted may or may not complete in the near future.

User or Operator Response: Explicitly retrieve the job from JES if and when the job completes execution.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

550 JesPutGet aborted, timeout exceeded

Severity: Permanent negative completion.

Explanation: A GET command was issued with a SITE FILETYPE=JES argument in effect. The GET command requested the output to be automatically retrieved. The output was not available to be retrieved within the time period allowed for the JOB to complete execution.

System Action: FTP server continues normal execution. The FTP client waits for the next command input. The job submitted may or may not complete in the near future.

User or Operator Response: Explicitly retrieve the job from JES if and when the job completes execution.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

550 JesPutGet terminated

Severity: Permanent negative completion.

Explanation: During the PUT or GET processing, a timer expired before the job completed.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Explicitly retrieve the job from JES.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesPutGet

---

550 Job name not found

Severity: Permanent negative completion.

Explanation: JES could not locate the JOBNAME specified in the command.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Job not cancelled, duplicate jobnames and no Jobid

Severity: Permanent negative completion.

Explanation: JES could not perform the requested function because more than one JOBNAME exists on which JES could perform the action.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Job not cancelled, job on output queue

Severity: Permanent negative completion.

Explanation: A DELETE (JES cancel) command was issued against a job that was executing. The job completed execution and all output was found to be on the output queue.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Jobid *&jobid.* **not found**

Severity: Permanent negative completion.

Explanation: JES could not locate the JOBID specified in the command.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Jobid with invalid syntax for subsystem

Severity: Permanent negative completion.

Explanation: The DELETE command issued did not conform to the required syntax.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Mismatched quotes on pathname *pathname*

Severity: Permanent negative completion.

Explanation: The pathname provided for a data transfer (retrieve or store), delete, or rename request was enclosed in a unbalanced set of single quotes.

System Action: The request is not performed.

User or Operator Response: Enclose the data set name in balanced quotes -- a quote at the beginning and the end of the name.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: delete, renamefr, rnto, retrieve, store_data

---

550 MKD failed. Dcbdsn data set *dcbdsn_name* **does not exist.**

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. The server could not find the model data set *dcbdsn_name* to determine the data set characteristics.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter. After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

550 MKD failed. Dcbdsn data set *dcbdsn_name* is not on a direct access volume.

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, the model data set *dcbdsn_name* was not on a direct access volume and therefore could not be used as a model data set.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter. After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Dcbdsn data set *dcbdsn_name* is a VSAM data set.

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, the model data set *dcbdsn_name* was a VSAM data set and therefore could not be used as a model for the data set characteristics.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter. After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Dcbdsn data set *dcbdsn_name* has an invalid dsorg.

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, the model data set *dcbdsn_name* was neither a partitioned data set or a physical sequential data set, and therefore could not be used as a model for the data set characteristics.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter. After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Dcbdsn data set *dcbdsn_name* has a recfm of *recfm* which is invalid for a PDS.

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, the model data set *dcbdsn_name* had a record format *recfm* that is not valid for a PDS. Record formats FBS, VBS, and VS are not valid record formats for a PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter, or a combination of DCBDSN and RECFM SITE parameters to use all of the characteristics of the model DCBDSN data set except the record format, which will be overridden by the RECFM parameter. After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Dcbdsn data set *dcbdsn_name* is migrated and noautorecall is specified.

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, the model data set *dcbdsn_name* was migrated and the server was currently in No AutoRecall mode, therefore the model data set could not be recalled to determine the data set characteristics.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE AUTORECALL command to allow the DCBDSN data set to be recalled, then reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Error locating dcbdsn data set *dcbdsn_name*

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN was in effect specifying that all new data sets be created with the same characteristics as the data set *dcbdsn_name*. However, an error occurred when the server issued the LOCATE macro to locate the model data set *dcbdsn_name*. The server was unable to determine the characteristics of the model data set *dcbdsn_name* and therefore could not allocate the new data set.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Use the SITE command to either specify a different DCBDSN data set, or to specify the data set characteristics without using the DCBDSN model data set parameter.

After changing the appropriate SITE parameters, reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Unit= *unit_name* **invalid for PDS.**

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS, but the current setting of the SITE UNIT parameter was not valid for a PDS (for example, the SITE UNIT was set to TAPE and a PDS cannot be allocated on a tape).

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Issue the SITE command with the UNIT parameter to change the setting of the UNIT parameter to a unit that is valid for a PDS, then reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Volume for dcbdsn data set *dcbdsn_name* **is not mounted and NoAutomount is specified.**

Severity: Permanent negative completion.

Explanation: The MKD command was specified to create a new PDS. The SITE DCBDSN parameter was in effect specifying that all newly created data sets be allocated using data set *dcbdsn_name* as a model data set for the data set characteristics. The volume containing *dcbdsn_name* was not currently mounted to the system, and the FTP server was currently in No AutoMount mode, so the FTP server could not get the volume mounted to the system to determine the data set characteristics of *dcbdsn_name* to use when allocating the new PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Issue the SITE AUTOMOUNT command to allow the volume containing *dcbdsn_name* to be mounted to the system, then reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Volume *volume* **is not mounted and NoAutoMount is specified.**

Severity: Permanent negative completion.

Explanation: The client issued the MKD command to create an MVS Partitioned Data Set at the server host. The SITE VOLUME command had previously been issued, or the VOLUME statement was specified in the FTP.DATA file, to specify that all new data sets be allocated on volume *volume*. Volume *volume* was not mounted to the system, and NoAutoMount had previously been specified to prevent the server from automatically requesting the volume to be mounted.

System Action: The MKD command is rejected. The FTP server waits for the next command from the client.

User or Operator Response: If possible, issue the SITE AUTOMOUNT subcommand to allow the FTP server to automatically request the operator to mount volume *volume.*. Otherwise, contact the MVS System Operator or MVS System Programmer to have the volume *volume* mounted to the system, then reissue the MKD subcommand.

System Programmer Response: If necessary, mount the volume *volume* to the system for use.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Volume= *serial* **is a tape and is invalid for a PDS.**

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE VOLUME parameter was in effect, specifying that all new data sets be allocated on volume *serial*. *serial* was determined to be a tape, and partitioned data sets cannot be allocated to tape.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Issue the SITE VOLUME command to change or reset the volume serial for new data sets to a direct access volume, then reissue the MKD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Error mounting dcbdsn data set *dcbdsn_name*

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN parameter was in effect, specifying that all newly allocated data sets should use the data set *dcbdsn_name* as a model data set for the data set characteristics. The volume containing data set *dcbdsn_name* was not currently mounted to the system, and when the FTP server attempted to get the volume mounted, an error occurred and the volume was unable to be mounted. The FTP server was therefore unable to determine the data set characteristics for the new PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Either contact the system programmer for the server system to get the volume containing *dcbdsn_name* mounted (note, the LIST *dcbdsn_name* command will display the volume that contains *dcbdsn_name*), or issue the SITE command to change the model DCB data set specified by the DCBDSN parameter, or use the other SITE parameters to specify the data set characteristics without using a model DCB, then reissue the MKD command.

System Programmer Response: Determine why the volume containing *dcbdsn_name* cannot be mounted to the system and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: mkd

550 MKD failed. Error retrieving dcbdsn data set *dcbdsn_name*

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE DCBDSN parameter was in effect, specifying that all newly allocated data sets should use the data set *dcbdsn_name* as a model data set for the data set characteristics. *dcbdsn_name* was migrated, and when the FTP server attempted to recall the data set, an error occurred and the data set was unable to be recalled. The FTP server was therefore unable to determine the data set characteristics for the new PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Either contact the system programmer for the server system to get the *dcbdsn__name* recalled, or issue the SITE command to change the model DCB data set specified by the DCBDSN parameter, or use the other SITE parameters to specify the data set characteristics without using a model DCB, then reissue the MKD command.

System Programmer Response: Determine why *dcbdsn_name* could not be recalled and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKD failed. Recfm *recfm* **is invalid for a PDS.**

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create a new PDS. The SITE RECFM parameter had a value of *recfm*, which is not a valid record format for a PDS.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Issue the SITE RECFM command to change the setting of RECFM to a record format that is valid for a PDS, then reissue the MKD command. (Incorrect record formats are FBS, VBS, and VS).

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 Mkd fails: *new_directory*. **User not authorized**

Severity: Permanent negative completion.

Explanation: The MKD command was issued to create the new PDS *new_directory*. However, some part of the high-level qualifiers for *new_directory* were protected by a security system, such as RACF, and the user did not have sufficient authority to create a data set by this name.

System Action: The MKD command is rejected. Control is returned to the client connection for further command processing.

User or Operator Response: Contact the security administrator to get the necessary authorization for creating the data set.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 MKDIR failed: *error*

Severity: Permanent negative completion.

Explanation: The client issued a MKD command to create an HFS directory at the FTP server host. The C runtime library function mkdir() was issued by the FTP server to create the directory, but the mkdir() function did not complete successfully. *error* is the error message returned by the C runtime library for the failing routine.

System Action: The MKD command is rejected. The server waits for the next command from the client.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPRD

Procedure Name: mkd

---

550 Name length error for pathname *pathname*

Severity: Permanent negative completion.

Explanation: The pathname provided for a data transfer (retrieve or store), delete, or rename request is too long. The name, when combined with the current working directory, must adhere to the following maximum lengths:

44 for a physical sequential data set
55 for a member of a PDS (includes the parentheses for the member name
1023 for a file in the hierarchical file system.

Note: The maximum length for a filename is 255 -- the total pathname maximum length is 1023.

System Action: The request is not performed.

User or Operator Response: Re-issue the request with a name that meets the limits for the type of data set or file.

System Programmer Response: None.

Source Data Set: EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: delete, renamefr, rnto, retrieve, store_data

---

550 No data sets found

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested a listing of one or more data sets, but the FTP server could not find any data sets that matched the requested path name.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Verify that the correct data set name or path was entered.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 No members found.

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested members of a partitioned data set, but no members were found in the PDS that matched the requested name.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Verify that the member name or path was entered correctly.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 No spool files available for *&jobid.*

Severity: Permanent negative completion.

Explanation: User attempted to retrieve spool file(s) from JES and no spool file(s) existed.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the command and re-issue.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 No spool files available for *&jobid.***, JesPutGet aborted**

Severity: Permanent negative completion.

Explanation: User attempted to retrieve spool file(s) from JES and no spool file(s) were available for processing.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the command and re-issue.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 Non-DASD data set *dsname* **cannot be processed.**

Severity: Permanent negative completion.

Explanation: The named data set does not have the correct data set organization. It is neither DASD nor tape.

System Action: The data transfer does not occur. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: file_check, find_file, find_unique_file

---

550 Open of *dsname* **failed.**

Severity: Permanent negative completion.

Explanation: The data set cannot be opened.

System Action: The data set is not sent. FTP continues.

User or Operator Response: None.

System Programmer Response: Look at the trace and find the following trace entry sequence:

```
seq_open_file: Ixy  -> r,recfm=* for dsname
seq_open_file: failed (aa): bbbb
```

The first line describes I/O mode as Input with mode values *x* and *y* that are defined as follows:

```
x = S --> stream I/O
x = R --> record I/O

y = B --> binary stream
y = T --> text stream
```

The second line describes error that occurred. *aa* is the errno value returned from the fopen of file. **bbbb** is the text associated with the errno value. The following describe values and text that could be observed:

(61): Error trying to define file
The file is already opened.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

550 Partitioned data set *'path name'* **does not exist.**

Severity: Permanent negative completion.

Explanation: The LIST or NLST command requested that members of a partitioned data set be listed, but the requested data set could not be found.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Verify that the PDS name was entered correctly.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 PASS command failed - getpwnam() error : *error*

Severity: Permanent negative completion.

Explanation: The client entered the USER and PASS commands to logon to the FTP server. While validating the user_id and password of the the client, the getpwnam() function issued by the FTP server to obtain the OMVS information about the user_id failed. *error* is the error message returned by the C runtime library for the getpwnam() function.

System Action: Logon to the FTP server is denied.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error indicated by *error*

Source Data Set: EZAFTPRA

Procedure Name: pass

---

550 Pathname does not contain valid Jobid

Severity: Permanent negative completion.

Explanation: The Jobid was incorrectly specified in the command.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct the request and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Rename fails: *dsname* **already exists.**

Severity: Permanent negative completion.

Explanation: The RNTO command was issued to rename a data set. A data set already exists with the new name.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: Either delete the existing data set or choose a different new name.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rnto

---

550 Rename fails: GDG name conversion failed.

Severity: Permanent negative completion.

Explanation: The relative name for a Generation Data Group (GDG) data set could not be converted into its absolute form.

System Action: The request is not performed.

User or Operator Response: Ensure that the name provided is the name of an existing GDG data set (for the old name) or that the new name specifies an exiting base name for a GDG.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rnto

---

550 Renaming attempt failed. Rc was *rc*.

Severity: Permanent negative completion.

Explanation: The RNTO command was issued to rename a data set. The rename was unsuccessful.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: None.

System Programmer Response: The return code *rc* is the errno value received when the rename C function was invoked to rename the data set.

Source Data Set: EZAFTPRF

Procedure Name: hfs_rnto, rnto

---

550 RETR fails: *dsname*. User not authorized.

Severity: Permanent negative completion.

Explanation: The RETR command was issued to read a data set. The requested data set is protected by a security system such as RACF, and the user is not authorized to read the data set.

System Action: The data set is not sent. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to read the data set.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

550 Retrieve of a whole Partitioned data set is not supported. Use MGET for this purpose.

Severity: Permanent negative completion.

Explanation: The data set that was named is a partitioned data set (PDS) and cannot be retrieved as a single file.

System Action: The data set is not sent. FTP continues.

User or Operator Response: To get all of the data set, retrieve all of the members of the data set. For example, use the MGET client command if available.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: file_check

---

550 RMD failed. Valid only for HFS files or partitioned data sets.

Severity: Permanent negative completion.

Explanation: The RMD command was issued to remove a directory, but the directory name entered was not a valid PDS or HFS directory.

System Action: FTP continues

User or Operator Response: Re-enter the command with the correct directory name.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

550 Rmd fails: *directory*. User not authorized

Severity: Permanent negative completion.

Explanation: The RMD command was issued to delete the partitioned data set *directory*, but *directory* was protected by a security system such as RACF and the user was not authorized to delete the data set.

System Action: The RMD command is rejected. Control returns to the client for further command processing.

User or Operator Response: Contact the owner of the data set to get the necessary authorization to delete the data set.

System Programmer Response: If required, authorize the user to delete the partitioned data set.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

550 RMDIR failed: *error*

Severity: Permanent negative completion.

Explanation: The client issued a RMD command to delete an HFS directory at the FTP server host. The C runtime library function rmdir() was issued by the FTP server to delete the directory, but the rmdir() function did not complete successfully. *error* is the error message returned by the C runtime library for the failing routine.

System Action: The RMD command is rejected. The server waits for the next command from the client.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer.

System Programmer Response: Correct the error indicated by *error*.

Source Data Set: EZAFTPRD

Procedure Name: rmd

---

550 RNFR fails: *dsname* does not exist.

Severity: Permanent negative completion.

Explanation: The RNFR command was issued to rename a data set. The rename was unsuccessful because the data set does not exist.

System Action: FTP continues.

User or Operator Response: Make sure that the correct data set name is entered and that the data set is catalogued.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: renamefr

550 RNFR fails: *dsname***. User not authorized.**

Severity: Permanent negative completion.

Explanation: The RNFR command was issued to rename a data set. The requested data set is protected by a security system such as RACF, and the user is not authorized to rename the data set.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to rename it.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: renamefr

---

550 RNTO fails: *dsname***. User not authorized.**

Severity: Permanent negative completion.

Explanation: The RNTO command was issued to rename a data set. The requested data set is protected by a security system such as RACF, and the user is not authorized to rename the data set.

System Action: The data set is not renamed. FTP continues.

User or Operator Response: Contact the owner of the data set for authorization to rename it.

System Programmer Response: None.

Source Data Set: EZAFTPRF

Procedure Name: rnto

---

550 SQL query not available. Can't load CAF routines.

Severity: Permanent negative completion.

Explanation: FTP was unable to load the CAF (Call Access Facility) modules it uses to provide support for SQL queries.

System Action: The command is rejected. FTP continues.

User or Operator Response: If FTP SQL support should be available, contact your system programmer. If FTP SQL support was not intended, issue a 'site filetype=' command to change the filetype from its current setting of 'SQL' to 'SEQ' or 'JES'.

System Programmer Response: If FTP SQL support is desired, ensure that the appropriate DSNLOAD library is included in the STEPLIB for the FTP server. (If FTP is started from the OE shell, the $STEPLIB environment variable must be set.) Restart the FTP server.

Source Data Set: EZAFTPMQ

Procedure Name: initSql

---

550 The user is not authorized to access the job

Severity: Permanent negative completion.

Explanation: A DELETE command was issued to a job the issuer does not have appropriate authority.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Correct and reissue the command. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose Problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 Transfer aborted

Severity: Permanent negative completion.

Explanation: This is the final reply of a group of replies.

System Action: FTP server continues normal execution. The FTP client waits for the next command input.

User or Operator Response: Refer to replies prior to the 550 reply.

System Programmer Response: None.

Source Data Set: EZAFTPMJ

Procedure Name: jesFileCheck, jesPutGet

---

550 Unable to append to *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred during the processing of the append request.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unable to append to fixed record format file in image mode

Severity: Permanent negative completion.

Explanation: The server does not supports an append request for a fixed format file when the data type is image.

System Action: The append is not performed.

User or Operator Response: Re-issue the request with data type ASCII or EBCDIC.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: append

---

550 Unable to append to member *name* **in Partitioned data set** *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred while trying to process a data set.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unable to append to the Null directory.

Severity: Permanent negative completion.

Explanation: The current working directory is the null directory. The server does not support the append command to the null directory.

System Action: The command is rejected. FTP continues.

User or Operator Response: Use the STOR or STOU command to store to the null directory or change the directory to use append.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

550 Unable to create data set *dsname* **for** *command* **command.**

Severity: Permanent negative completion.

Explanation: The server attempted to allocate to a new data set to store data. The dynamic allocation was unsuccessful. *command* is STOR, STOU, or APPE.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *seq_create_file*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unable to delete *directory* **(Error code:** *error code***, Reason code:** *reason code***)**

Severity: Permanent negative completion.

Explanation: The RMD or DELE command was issued to delete PDS *directory*, but when the FTP server attempted to delete the data set, the delete did not complete successfully. Possible causes for this error include the data set is currently in use by someone else, the data set has an unexpired retention period, the data set is located on a tape volume, or the direct access volume containing the data set is not writable.

System Action: The RMD or DELE command is rejected. Control returns to the client for further command processing.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer for the server system.

System Programmer Response: The *error code* and *reason code* displayed contain the S99ERROR and S99INFO fields from DYNALLOC. Refer to the Dynamic Allocation return code section in the "OS/390 Programming: Authorized Assembler Services Guide" to determine why the data set cannot be deleted and correct the problem.

Source Data Set: EZAFTPRD, EZAFTPRF

Procedure Name: rmd, delete

---

550 Unable to delete *dsname* **(Rc =** *rc***)**

Severity: Permanent negative completion.

Explanation: The attempt to delete an HFS file was unsuccessful.

System Action: The file is not deleted. FTP continues.

User or Operator Response: None.

System Programmer Response: The return code *rc* is the errno value received when the remove C function was invoked to delete the file.

Source Data Set: EZAFTPRF

Procedure Name: hfs_delete

---

550 Unable to obtain data set list

Severity: Permanent negative completion.

Explanation: A service call was issued to obtain the list of requested data sets from the MVS system, but the service call was unsuccessful.

System Action: The LIST or NLST command is rejected.

User or Operator Response: Retry the command. If the problem persists, contact the system programmer for the FTP server system.

System Programmer Response: If necessary, re-create the problem with FTP server traces turned on. Locate the last occurrence of the trace message “list2: return code *rc* from gtlist.”. This trace message will contain the return code from the service call that was unsuccessful. Determine the cause and correct the problem.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 Unable to perform LIST/NLST command due to error with /dev/null.

Severity: Permanent negative completion.

Explanation: During server initialization, the server attempted to open STDOUT and STDERR to the /dev/null character special file. It was unable to do so, and the server is not able to perform the LIST or NLST subcommand for HFS files.

System Action: The command is rejected. FTP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Error Message EZYFT48E logged during server initialization will contain the reason for the /dev/null failure. Correct this error and restart the FTP server.

Source Data Set: EZAFTPRD

Procedure Name: list2

---

550 Unable to send *dsname*

Severity: Permanent negative completion.

Explanation: The named data set is for a generation data group (GDG) that does not exist.

System Action: The data set is not sent. FTP continues.

User or Operator Response: Insure that the GDG exists.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: file_check, retrieve

---

550 Unable to store *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred during the processing of the store request.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

550 Unable to store member *name* **in Partitioned data set** *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred during the processing of the store request.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unable to store unique *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred during the processing of the store unique request.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unable to store unique member *name* **in Partitioned data set** *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred while trying to process a data set.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: create_msg

---

550 Unexpected status for *dsname*

Severity: Permanent negative completion.

Explanation: An internal error occurred while trying to find a data set.

System Action: The request is not performed.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Contact the IBM Support Center.

Source Data Set: EZAFTPRS

Procedure Name: find_file, find_unique_file

---

550 Unknown HOME directory for pathname *path*

Severity: Permanent negative completion.

Explanation: A command was issued that required the server to resolve the user's HOME directory name (for example, a pathname was entered that began with the directory notation ~/) but the server was unable to determine the user's HOME directory.

System Action: The command is rejected. FTP continues.

User or Operator Response: Verify that the OMVS user_id has a HOME directory. If necessary, define a HOME directory for the user_id.

System Programmer Response: None.

Source Data Set: EZAFTPMJ, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: jesPutGet, rnto, delete, renamefr, retrieve, store_data

---

550 Unknown return code from Get JES spool request: *number*

Severity: Permanent negative completion.

Explanation: During the processing of a JES request, an unknown return code was returned from JES.

System Action: Unpredictable results may occur.

User or Operator Response: Issue the command again. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 Unknown return code from Get JES spool request: *number***, JesPutGet aborted**

Severity: Permanent negative completion.

Explanation: During the processing of a JES request, an unknown return code was returned from JES.

System Action: Unpredictable results may occur.

User or Operator Response: Issue the request again. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesGet

---

550 Unknown return code from JES cancel request

Severity: Permanent negative completion.

Explanation: A request was made to delete a job from JES. During the processing of this request, JES provided an unknown return code.

System Action: Unpredictable results may occur.

User or Operator Response: Issue the request again. If problem persists, contact appropriate support personnel.

System Programmer Response: Diagnose problem.

Source Data Set: EZAFTPMJ

Procedure Name: jesDelete

---

550 User Exit refuses this Job to be submitted by *userid*

Severity: Informational

Explanation: The job submitted by the indicated user ID has been denied. The FTCHJES routine is part of the FTP server security user exits. This routine is called if the server is in filetype =jes mode and the client tries to submit a job. The exit can allow or refuse the job to be submitted to the JES internal reader.

System Action: TCPIP continues.

User or Operator Response: Contact the System Programmer to obtain access to this command.

System Programmer Response: None.

Source Data Set: EZAFTSMJ

Procedure Name: WriteIt

550 Volume containing *dsname* is not mounted and NoAutoMount specified.

Severity: Permanent negative completion.

Explanation: The command requested MVS data set *dsname*, but the volume containing *dsname* was not mounted to the system and the FTP server was currently in No Automount mode.

System Action: The command is rejected.

User or Operator Response: If possible, issue the SITE AUTOMOUNT command to allow the volume containing the data set to be automatically mounted; otherwise, contact the system operator to mount the volume to the system. Then reissue the command.

System Programmer Response: None.

Source Data Set: EZAFTPRD, EZAFTPRF, EZAFTPRR, EZAFTPRS

Procedure Name: list2, delete, renamefr, file_check, find_file, find_unique_file

---

550-volume for *"new_directory"* is not mounted and NoAutomount is specified.

Severity: Permanent negative completion.

Explanation: The volume containing the requested *new_directory* was not mounted and the server was currently in No AutoMount mode, therefore the volume could not be mounted to determine whether the data set for the *new_directory* was a PDS.

System Action: The CWD command is rejected.

User or Operator Response: Issue the SITE AUTOMOUNT command to allow the volume to be mounted, then reissue the CWD command.

System Programmer Response: None.

Source Data Set: EZAFTPRD

Procedure Name: cwd2, list2

---

550 Volume is not ready and automatic tape mounts are not allowed.

Severity: Permanent negative completion.

Explanation: A request to retrieve a data set from or store a data set on a tape volume was received. The tape volume is not mounted and the end user has requested NoAutoTapeMount.

System Action: The request is not performed.

User or Operator Response: Request that the tape be mounted by the operator and then re-issue the request. Also, the SITE AUTOTAPEMOUNT subcommand can be issued to allow automatic tape mounts to occur.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

---

550 Volume *name* is not mounted and NoAutoMount is specified.

Severity: Permanent negative completion.

Explanation: A store request was received for which a new data set must be created. The volume on which the data set was to be created is not mounted and the server was currently in NoAutoMount mode.

System Action: The request is not performed.

User or Operator Response: Request that the volume be mounted by the operator and then re-issue the request. Also, the SITE AUTOMOUNT subcommand can be issued to allow automatic DASD mounts to occur.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

---

550 VSAM data set *dsname* cannot be processed.

Severity: Permanent negative completion.

Explanation: The named data set is a VSAM data set and cannot be processed.

System Action: The data set is not sent. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: file_check, find_File, find_unique_file

---

551 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS "kill" command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

551 Transfer aborted: attempt to connect to *db2name* failed *(code)*

Severity: Permanent negative completion.

Explanation: FTP attempted to process a SQL request, but was unable to connect to the DB2 subsystem named *db2name*.

System Action: No data is sent. FTP continues.

User or Operator Response: If the DB2 subsystem name *(db2name)* is incorrect, issue a 'site db2=' command for the correct DB2 subsystem name, and resubmit the SQL query. If the DB2 subsystem name is correct, contact your system programmer for FTP.

System Programmer Response: Ensure that the DB2 subsystem has been started. See the *DB2 Messages and Codes* for a detailed explanation of *code*.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

551 Transfer aborted: SQL CURSOR DECLARE failure

Severity: Permanent negative completion.

Explanation: DB2 encountered an error while processing a SQL statement submitted through FTP. The FTP output file contains the error message from DB2.

System Action: FTP continues.

User or Operator Response: If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact your system programmer for FTP.

System Programmer Response: Correct the error indicated by the message contained in the output file. For detailed explanation of DB2 error codes, see the *DB2 Messages and Codes*.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

551 Transfer aborted: SQL CURSOR OPEN failure

Severity: Permanent negative completion.

Explanation: DB2 encountered an error while processing a SQL statement submitted through FTP. The FTP output file contains the error message from DB2.

System Action: FTP continues.

User or Operator Response: If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact your system programmer for FTP.

System Programmer Response: Correct the error indicated by the message contained in the output file. For detailed explanation of DB2 error codes, see the *DB2 Messages and Codes*.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

551 Transfer aborted: SQL not available. Attempt to open plan *plan name***failed** *(code)*

Severity: Permanent negative completion.

Explanation: FTP attempted to process a SQL request, but was unable to open FTP's DB2 subsystem plan named *plan name*

System Action: No data is sent. FTP continues.

User or Operator Response: If FTP SQL support should be available, contact your system programmer for FTP. If a SQL query was not intended, issue 'site filetype=' command to change the filetype from 'SQL' to 'SEQ' or 'JES' before the next FTP retrieve command.

System Programmer Response: Ensure that a BIND was done for plan *plan name* and that execute authorization was granted for the plan.

Note: If you used a plan name other than EZAFTPMQ for the EZAFTPMQ DBRM, your FTP.DATA file must contain a DB2PLAN statement to specify the plan name that the FTP server is to use. See *DB2 Messages and Codes* for a detailed explanation of *code*.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

551 Transfer aborted: SQL PREPARE/DESCRIBE failure

Severity: Permanent negative completion.

Explanation: DB2 encountered an error while processing a SQL statement submitted through FTP. The FTP output file contains the error message from DB2.

System Action: FTP continues.

User or Operator Response: Correct the error indicated in the output file, and resubmit the SQL request.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

---

554 command terminated due to server shutdown in progress

Severity: Permanent negative completion.

Explanation: The FTP server was processing a subcommand from the client when the process was stopped either by an OMVS “kill” command, or by a server abend. The server has begun shutdown processing.

System Action: The subcommand in progress is terminated. The FTP server session is terminated. The FTP server session process is terminated.

User or Operator Response: None.

System Programmer Response: If the server process was stopped due to an abend, contact the IBM Support Center.

Source Data Set: EZAFTPHC

Procedure Name: pgmstpc, pgmabndc

---

554 Requested action not taken: file not found.

Severity: Completion.

Explanation: A restart request was received for the FTP server to resume storing into a data set or file at a point indicated by a restart marker. The data set or file does not exist so the restart is not possible.

System Action: FTP continues.

User or Operator Response:

System Programmer Response:

Source Data Set: EZAFTPRS

Procedure Name: find_hfs_File, store_data

---

554 Requested action not taken: GDG name conversion failed.

Severity: Completion.

Explanation: A request was received to store a relative generation number of a Generation Data Group (GDG) data set. The relative name could not be converted into an absolute name.

System Action: FTP continues.

User or Operator Response: Ensure that the data set name is a valid generation data group.

System Programmer Response: None.

Source Data Set: EZAFTPRS

Procedure Name: store_data

554 Requested action not taken: invalid REST parameter.

Severity: Permanent negative completion.

Explanation: A RETR command is being processed as part of a restart of a checkpointed command. Before the RETR was sent, a REST command was processed and a restart marker was saved. The restart marker is incorrect.

System Action: The RETR command is rejected.

User or Operator Response: None.

System Programmer Response: Look at the trace to determine the error that occurred when the repositioning was done.

Source Data Set: EZAFTPRR, EZAFTPRS

Procedure Name: retrieve, store_data

554 Transfer aborted: unsupported SQL statement

Severity: Permanent negative completion.

Explanation: FTP has retrieved a file (or data set) while filetype = "SQL' but the contents of the file contain a SQL statement that is not supported by FTP.

System Action: The RETR command is rejected.

User or Operator Response: If a SQL query was intended, reissue the RETR command for a file that contains a SQL SELECT statement. If SQL query was not intended, issue a 'site filetype=' command to change the current setting of filetype to either 'SEQ' or 'JES'.

System Programmer Response: None.

Source Data Set: EZAFTPRR

Procedure Name: retrieve

Appendix B. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by Sun Microsystems, Massachusetts Institute of Technology, Digital Equipment Corporation, and The University of California.

Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts. All Rights Reserved.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
AD/Cycle
AIX
AIX/ESA
BookManager
C/370
CICS
DB2
DFSMS
DFSMS/MVS
ESCON
ES/9000
EtherStreamer
Extended Services
GDDM
Hardware Configuration Definition
IBM
LANStreamer
Library Reader
MVS/ESA
MVS/SP
MVS/XA
NetView
OpenEdition
OS/2
OS/390
PS/2
RACF
RISC System/6000
RS/6000
SAA
S/390
System/370
System/390
TURBOWAYS
VTAM
3090

The following terms are trademarks of other companies:

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Bibliography

This bibliography lists the publications for IBM TCP/IP products.

IBM TCP/IP Publications

The following sections describe the books associated with IBM TCP/IP products.

OS/390 TCP/IP OpenEdition: Publications

- *OS/390 TCP/IP OpenEdition Configuration Guide*, SC31-8304.

  This book is for people who want to configure, customize, administer, and maintain OS/390 TCP/IP OpenEdition: . Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition Diagnosis Guide*, SC31-8492.

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the IBM OS/390 TCP/IP OpenEdition: product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *OS/390 TCP/IP OpenEdition Messages and Codes*, SC31-8307.

  This book explains the informational and error messages issued by IBM OS/390 TCP/IP OpenEdition: . It can help users, operators, or system programmers to diagnose and fix problems identified by OS/390 TCP/IP OpenEdition: error messages.

- *OS/390 TCP/IP OpenEdition Planning and Release Guide*, SC31-8303.

  This book is intended to help you plan for OS/390 TCP/IP OpenEdition: for MVS whether you are migrating from a previous version or installing TCP/IP for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with OS/390 TCP/IP OpenEdition: .

- *OS/390 TCP/IP OpenEdition Programmer's Reference*, SC31-8308.

  This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing.

  This book is for people who want to use the supplied interfaces while writing application programs that access OS/390 TCP/IP OpenEdition: . Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition User's Guide*, GC31-8305.

  This book is for people who want to use OS/390 TCP/IP OpenEdition: for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for MVS Publications

- *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, SC31-8069

  This book explains how to plan for, install, customize, and use the OpenEdition MVS Applications Feature. The Feature consists of applications and interfaces for direct access to the OpenEdition MVS environment. For example, users of the Feature can use MVS, UNIX, or AIX commands to transfer files, log in to the OpenEdition environment without going through TSO, and run commands remotely. This book also explains how to improve performance and diagnose problems when using the Feature.

- *TCP/IP for MVS: Application Programming Interface Reference*, SC31-7187

  This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

- *TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*, SC31-7131

  This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

- *TCP/IP for MVS: Customization and Administration Guide*, SC31-7134.

  This book is for people who want to configure, customize, administer, and maintain TCP/IP for MVS. Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: Diagnosis Guide*, LY43-0105

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the IBM TCP/IP for MVS product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference*, SC31-7186

This book is for programmers who want to write application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

- *TCP/IP for MVS: Messages and Codes*, SC31-7132.

  This book explains the informational and error messages issued by IBM TCP/IP for MVS. It can help users, operators, or system programmers to diagnose and fix problems identified by TCP/IP for MVS error messages.

- *TCP/IP for MVS: Network Print Facility*, SC31-8074

  This book is for system programmers and network administrators who need to prepare their network to route VTAM, JES2, or JES3 printer output to remote printers using TCP/IP for MVS.

- *TCP/IP for MVS: Offloading TCP/IP Processing*, SC31-7133

  This book is for people who want to install and configure the Offload feature on IBM 3172 Model 3 Interconnect Controllers. This book is also for people who want to use and customize the Offload feature of TCP/IP for MVS.

- *TCP/IP for MVS: Planning and Migration Guide*, SC31-7189

  This book is intended to help you plan for TCP/IP for MVS whether you are migrating from a previous version or installing TCP/IP for MVS for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with TCP/IP for MVS.

- *TCP/IP: Performance Tuning Guide*, SC31-7188

  This book describes how to improve the performance of your network operations.

- *TCP/IP for MVS: Programmer's Reference*, SC31-7135

  This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing.

  This book is for people who want to use the supplied interfaces while writing application programs that access TCP/IP for MVS. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: User's Guide*, SC31-7136

  This book is for people who want to use TCP/IP for MVS for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for VM Publications

The following list describes books in the IBM TCP/IP for VM library.

- *IBM TCP/IP Version 2 Release 4 for VM: Messages and Codes*, SC31-6151.

  This book is for system programmers who want to diagnose and fix problems identified by TCP/IP for VM error messages.

- *IBM TCP/IP Version 2 Release 4 for VM: Planning and Customization*, SC31-6082.

  This book is for system programmers who want to plan and customize the TCP/IP for VM environment.

- *IBM TCP/IP Version 2 Release 4 for VM: Programmer's Reference*, SC31-6084.

  This book is for application and system programmers who want to write application programs that use TCP/IP for VM. Application programmers should know the VM operating system.

- *IBM TCP/IP Version 2 Release 4 for VM: User's Guide*, SC31-6081.

  This book is for people who want to use TCP/IP for VM for data communication. Familiarity with VM operating system, IBM Command Processor (CP), and IBM Conversational Monitor System (CMS) is recommended.

TCP/IP for OS/2 Publication

IBM TCP/IP Version 3.0 for OS/2: Programmer's Reference, SC31-6077.

This book provides application and system programmers with the information required to write application programs that use TCP/IP for OS/2. Programmers should know the OS/2 operating system.

TCP/IP for DOS Publications

The following list describes books in the IBM TCP/IP for DOS library.

- *IBM TCP/IP Version 2.1.1 for DOS: Command Reference*, SX75-0083.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, understand DOS operating system concepts, and be familiar with the *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*.

- *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration*, SC31-7047.

  This book provides system programmers, network administrators, and workstation users responsible for installing TCP/IP for DOS with the information required to plan and implement the installation of TCP/IP for DOS. The topics include hardware and software

requirements, pre-installation system performance considerations, instructions for installing TCP/IP for DOS, instructions for customizing the TCP/IP for DOS environment, and installation examples.

- *IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference*, SC31-7046.

  This book is for application and system programmers to aid them in writing application programs that use TCP/IP for DOS on a workstation. Application programmers should know the DOS operating system and multitasking operating system concepts. Application programmers should be knowledgeable in the C programming language.

- *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*, SC31-7045.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, and also understand DOS operating system concepts.

TCP/IP for AIX (RS/6000, PS/2, RT, 370) Publications

The following list shows books in the TCP/IP for AIX library.

- *AIX Operating System TCP/IP User's Guide*, SC23-2309.
- *AIX PS/2 TCP/IP User's Guide*, SC23-2047.
- *TCP/IP for IBM X-Windows on DOS 2.1*, SC23-2349.

TCP/IP for AS/400 Publications

The following list shows books in the TCP/IP for AS/400 library.

- *IBM AS/400 Communications: TCP/IP Guide*, SC41-9875.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.

Other IBM TCP/IP Publications

The following list shows other available IBM TCP/IP books.

- *IBM Local Area Network Technical Reference*, SC30-3383.
- *IBM TCP/IP for VM and MVS: Diagnosis Guide*, LY43-0013.
- *TCP/IP and National Language Support*, GG24-3840.
- *TCP/IP Introduction*, GC31-6080.
- *TCP/IP Tutorial and Technical Overview*, GG24-3376.

IBM Operating System Publications

The following lists show books about various IBM operating systems.

AIX Publications

- *AIX Communications Concepts and Procedures for IBM RISC System/6000*, GC23-2203.
- *AIX Communications Programming Concepts*, SC23-2206.
- *IBM AIX Operating System Technical Reference, Volume 1*, SC23-2300.
- *IBM AIX Operating System Technical Reference, Volume 2*, SC23-2301.

AS/400 Publications

- *IBM AS/400 CL Reference Manual Volume 1*, SC21-9775.
- *IBM AS/400 CL Reference Manual Volume 2*, SC21-9776.
- *IBM AS/400 CL Reference Manual Volume 3*, SC21-9777.
- *IBM AS/400 CL Reference Manual Volume 4*, SC21-9778.
- *IBM AS/400 CL Reference Manual Volume 5*, SC21-9779.
- *IBM AS/400 Communications: APPN Network User's Guide*, SC21-8188.
- *IBM AS/400 Communications: Programmer's Guide*, SC21-9590.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.
- *IBM AS/400 Device Configuration Guide*, SC21-8106.
- *IBM AS/400 Programming: Command Reference Summary*, SC21-8076.
- *IBM AS/400 Programming: Data Management Guide*, SC21-9658.
- *IBM AS/400 System Operations: Database Coordinator' Guide*, SC21-8086.
- *IBM AS/400 System Operations: Operator's Guide*, SC21-8082.

DOS Publications

- *DOS Getting Started Version 5.00*, SA40-0637.
- *DOS 5.02 Technical Reference*, S16G-4559.
- *DOS/Windows Client Getting Started*, SC09-3000.
- *PC DOS 6.1 Command Reference*, S71G-3634.

MVS Publications

For a complete description of the library for MVS/ESA Version 5, see *OS/390 Information Roadmap*, GC28-1727. See also "JES Publications" on page 247.

OS/2 Publications

- *IBM OS/2 Warp Server Up and Running!*, S25H-8004
- *IBM Official Guide to Using OS/2 Warp*, ISBN 1-56884-466-2 (Karla Stagray and Linda S. Rogers; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)
- *IBM OS/2 Warp Internet Connection: Your Key to Cruising the Internet and the World Wide Web*, ISBN 1-56884-465-4 (Deborah Morrison; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)

OS/390 Publications

- *OS/390 Information Roadmap*, GC28-1727.

  This book describes the documentation for the specific elements included in OS/390.
- *OS/390 Planning for Installation Release 3*, GC28-1726.

  This book is intended to help you plan for the installation of OS/390. It describes migration, installation, hardware and software requirements, and coexistence considerations.
- *OS/390 OpenEdition Introduction*, GC28-1889.
- *OS/390 OpenEdition Planning*, SC28-1890.
- *OS/390 OpenEdition User's Guide*, SC28-1891.
- *OS/390 OpenEdition Command Reference*, SC28-1892.
- *OS/390 OpenEdition Messages and Codes*, SC28-1908.
- *OS/390 Language Environment Programming Guide*, SC28-1939.
- *OS/390 Language Environment Programming Reference*, SC28-1940.
- *OS/390 OpenEdition Programming: Assembler Callable Services Reference*, GC23-3870.
- *OS/390 Open Systems Adapter Support Facility User's Guide*, SC28-1855.
- *Planning for the System/390 Open Systems Adapter Feature*, GC23-3870.

VM Publications

- *VM/ESA CMS Command Reference Summary*, SX24-5249.
- *VM/ESA CP Planning and Administration for 370*, SC24-5430.
- *VM/ESA CP Programming Services for 370*, SC24-5435.
- *VM/ESA Group Control System Reference for 370*, SC24-5426.
- *VM/ESA: Library Guide and Master Index*, GC23-0367.
- *VM/ESA: Master Index for 370*, GC24-5436.
- *VM/ESA Service Introduction and Reference*, SC24-5444.
- *VM/SP CMS Command Reference*, ST00-1981.
- *VM/SP Group Control System Macro Reference*, SC24-5250.
- *VM/SP Installation Guide*, SC24-5237.
- *VM/SP High Performance Option: Library Guide and Master Index*, GC23-0187.
- *VM/SP System Facilities for Programming*, SC24-5288.
- *VM/XA CP Programming Services*, SC23-0370.
- *VM/XA Diagnosis Reference*, LY27-8054.
- *VM/XA Installation and Service*, SC23-0364.
- *VM/XA SP Group Control System Command and Macro Reference*, SC23-0433.

IBM Software Publications

The following sections describe the books associated with IBM software products.

ACF/VTAM Publications

The following list shows books in the VTAM Version 4 Release 4 library.

- *VTAM Installation and Migration Guide*, GC31-8367.
- *VTAM Release Guide*, GC31-6545.
- *VTAM Network Implementation Guide*, SC31-8370.
- *VTAM Resource Definition Reference*, SC31-8377.
- *VTAM Resource Definition Samples*, SC31-8378.
- *VTAM Customization*, LY43-0075.
- *VTAM V4R1 Operation*, SC31-8372.
- *VTAM Messages*, GC31-8368.
- *VTAM Codes*, GC31-8369
- *VTAM V4R1 Programming*, SC31-8373.
- *VTAM Guide to Programming for LU 6.2*, SC31-8374.
- *VTAM Programming Reference for LU 6.2*, SC31-8375.
- *VTAM Programming for CSM*, SC31-8420.
- *VTAM CMIP Services and Topology Agent Programming Guide*, SC31-8365.
- *VTAM Diagnosis*, LY43-0078.
- *VTAM Data Areas for MVS/ESA Volume 1*, LY43-0076.
- *VTAM Data Areas for MVS/ESA Volume 2*, LY40-0077.
- *APPC Application Suite User's Guide*, SC31-6532.
- *APPC Application Suite Administration*, SC31-6533.
- *APPC Application Suite Programming*, SC31-6534.
- *VTAM AnyNet Guide to Sockets over SNA*, SC31-8371.
- *VTAM AnyNet Guide to SNA over TCP/IP*, SC31-8376.
- *VTAM Glossary*, GC31-8366.

- *Planning for NetView, NCP, and VTAM*, SC31-8063
- *Planning for Integrated Networks*, SC31-8062
- *VTAM Licensed Program Specifications*, GC31-8379.
- *VTAM Operation Quick Reference*, SX75-0208.

DATABASE 2 Publications

The following lists show books in the DATABASE 2 library.

DATABASE 2 Version 2

- *IBM DATABASE 2 Version 2: Administration Guide*, SC26-4374.
- *IBM DATABASE 2 Version 2: Application Programming and SQL Guide*, SC26-4377.
- *IBM DATABASE 2 Version 2: Messages and Codes*, SC26-4379.
- *IBM DATABASE 2 Version 2: Reference Summary*, SX26-3771.
- *IBM DATABASE 2 Version 2: SQL Reference*, SC26-4380.

DATABASE 2 Version 3

- *IBM DATABASE 2 Version 3: DB2 Administration Guide*, SC26-4888.
- *IBM DATABASE 2 Version 3: DB2 Application Programming and SQL Guide*, SC26-4889.
- *IBM DATABASE 2 Version 3: DB2 Messages and Codes*, SC26-4892.
- *IBM DATABASE 2 Version 3: DB2 Reference Summary*, SX26-3801.
- *IBM DATABASE 2 Version 3: DB2 SQL Reference*, SC26-4890.

ISPF Publication

ISPF Dialog Management Guide and Reference, SC34-4266.

JES Publications

- *MVS/ESA Library Guide with JES2*, SC28-1423.
- *MVS/ESA Library Guide with JES3*, SC28-1424.

MVS/DFP Publications

- *MVS/DFP Version 3 Release 3: Customizing and Operating the Network File System Server*, SC26-4832.
- *MVS/DFP Version 3 Release 3: Macro Instructions for Data Sets*, S26-4747.
- *MVS/DFP Version 3 Release 3: Using Data Sets*, SC26-4749.
- *MVS/DFP Version 3 Release 3: Using the Network File System Server*, SC26-4732.

Network Control Program (NCP) Publications

- *ACF/NCP V7R1 IP Router Planning and Installation Guide*, GG24-3974.
- *NCP and EP Reference*, LY43-0029.
- *NCP, SSP, and EP Generation and Loading Guide*, SC31-6221.
- *NCP, SSP, and EP Resource Definition Guide*, SC31-6223.
- *NCP, SSP, and EP Resource Definition Reference*, SC31-6224.

TME 10 NetView for OS/390 Publications

For a complete listing of TME 10 NetView for OS/390 Publications, refer to the *TME 10 NetView for OS/390 Library Reference,* SC31-8249.

Networking Systems Cross-Product Library

The following list shows books in the Networking Systems cross-product library.

- *Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM*, SX75-0092.
- *Planning for Integrated Networks*, SC31-8062.
- *Planning for NetView, NCP, and VTAM*, SC31-8063.

OpenEdition MVS Publications

The following list shows selected books in the OpenEdition MVS library.

- *OS/390 OpenEdition Introduction*, GC28-1889
- *OS/390 OpenEdition Planning*, SC28-1890

Programming Publications

The following list shows books about various programming applications.

- *IBM C/370 Diagnosis Guide and Reference*, LY09-1804 (feature 8082).
- *IBM C/370 General Information Manual*, GC09-1386.
- *IBM C/370 Installation and Customization Guide Version 2 Release 1.0*, GC09-1387.
- *IBM C/370 Programming Guide*, SC09-1384.
- *IBM C/370 Reference Summary*, SX09-1211.
- *IBM C/370 User's Guide*, SC09-1264.
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663.
- *IBM TSO Extensions CLISTs*, SC28-1876.
- *IBM TSO Extensions Command Language Reference*, GX23-0015.

- *IBM TSO Extensions Interactive Data Transmission Facility: User's Guide*, SC28-1104.
- *IMS/ESA V3R1 Application Programming: DL/I Calls*, SC26-4274.
- *HiPPI User's Guide and Programmer's Reference*, SA23-0369.
- *Parallel I/O Access Methods Programmer's Guide*, SC26-4648.
- *VS Pascal Application Programming Guide*, SC26-4319.
- *VS Pascal Diagnosis Guide and Reference*, LY27-9525.
- *VS Pascal General Information*, GT00-2664.
- *VS Pascal Installation and Customization for MVS*, SC26-4321.
- *VS Pascal Installation and Customization for VM*, SC26-4342.
- *VS Pascal Language Reference*, SC26-4320.

RACF Publications

The following list shows books in the RACF library.

- *IBM Resource Access Control Facility (RACF): General Information Manual*, GT00-2820.
- *IBM Resource Access Control Facility (RACF): User's Guide*, SC28-1341.
- *External Security Interface (RACROUTE) Macro Reference*, GC28-1366.
- *RACF Publications Order Guide*, GX22-0002.
- *Resource Access Control Facility (RACF) Security Administrator's Guide*, SC28-1340.
- *System Programming Library: RACF*, SC28-1343.

SMP/E Publications

The following list shows books in the SMP/E Release 8 library.

- *SMP/E Diagnosis Guide*, SC23-3130.
- *SMP/E Messages and Codes*, SC28-1107.
- *SMP/E Reference*, SC28-1107.
- *SMP/E Reference Summary*, SX22-0006.
- *SMP/E User's Guide*, SC28-1302.

VSAM Publication

MVS/370 VSAM Administration Guide, GC26-4066.

X.25 NPSI Publication

- *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning Version 3*, LY30-5610.
- *X.25 Network Control Program Packet Switching Interface Host Programming*, SC30-3502.
- *X.25 Network Control Program Packet Switching Interface Planning and Installation*, SC30-3470.

IBM Hardware Publications

The following sections describe the books associated with IBM hardware products.

System/370 and System/390 Publications

The following list shows the principles of operation manuals for the System/370 and System/390 processors.

- *IBM ESA/370 Principles of Operation*, SA22-7200.
- *IBM ESA/390 Principles of Operation*, SA22-7201.
- *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085.
- *IBM System/370 Principles of Operation*, GA22-7000.
- *S/360, S/370, and S/390 I/O Interface Channel to Channel Control Unit OEMI*, GA22-6974.

3172 Interconnect Controller Publications

The following list shows books in the IBM 3172 Interconnect Controller library.

- *IBM Interconnect Controller Program User's Guide*, SC30-3525.
- *IBM 3172 Interconnect Controller Installation and Service Guide*, GA27-3861.
- *IBM 3172 Interconnect Controller Operator's Guide*, GA27-3860.
- *IBM 3172 Interconnect Controller Planning Guide*, GA27-3867.
- *IBM 3172 Interconnect Controller Status Codes*, GA27-3951.

3270 Information Display System Publication

3270 Information Display System: 3270 Data Stream Programmer's Reference, GA23-0059.

8232 LAN Channel Station Publications

The following list shows books in the IBM 8232 LAN Channel Station library.

- *IBM LAN Channel Support Program: Version 1.0 User's Guide*, SC30-3458.
- *IBM 8232 LAN Channel Station: Installation and Testing*, GA27-3796.
- *IBM 8232 LAN Channel Station: Operating Guide*, GA27-3785.

9370 Publications

The following list shows books in the 9370 library.

- *IBM 9370 Information System: Using the X.25 Communications Subsystem*, SA09-1742.
- *IBM 9370 Information System X.25 Communications Subsystem Description*, SA09-1743.
- *VM/ESA: Connectivity Planning, Administration, and Operation Release 1*, SC24-5448.

Other TCP/IP-Related Publications

The following sections describe other books associated with TCP/IP.

- *The Art of Distributed Application: Programming Techniques for Remote Procedure Calls*, John R. Corbin, Springer-Verlog, 1991.
- *CAE Specification: X/Open Transport Interface (XTI),* X/Open Company Ltd., U. K., 1992, SC31-8005.
- *IEEE Network Magazine*, July 1990.
- *TCP/IP Illustrated Volume I: The Protocols*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1994, SR28-5586.
- *TCP/IP Illustrated Volume II: The Implementation*, Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1995, SR28-5630.
- *TCP/IP Illustrated Volume III*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1996, SR23-7289
- *Interoperability Report*, Volume 3, No. 3, March 1989.
- "MIB II Extends SNMP Interoperability," C. Vanderberg, *Data Communications*, October 1990.
- *"Network Management and the Design of SNMP,"* J.D. Case, J.R. Davin, M.S. Fedor, M.L. Schoffstall.
- *"Network Management of TCP/IP Networks: Present and Future,"* A. Ben-Artzi, A. Chandna, V. Warrier.
- *The Simple Book: An Introduction to Management of TCP/IP-based Internets*, Marshall T Rose, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- "Special Issue: Network Management and Network Security," *ConneXions-The Interoperability Report*, Volume 4, No. 8, August 1990.
- *UNIX Programmer's Reference Manual*, (4.3 Berkeley Software Distribution, Virtual VAX-11 Version). Department of Electrical Engineering and Computer Science. University of California, Berkeley, 1988.

OSF/Motif Publications

The following list shows OSF/Motif books.

- *OSF/Motif Application Environment Specifications*, (AES), Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640483-9.
- *OSF/Motif Programmer's Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640509-6.
- *OSF/Motif Programmer's Reference*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640517-7.
- *OSF/Motif Style Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640491-X.
- *OSF/Motif User's Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640525-8.

Sun (RPC) Publications

The following list shows Sun Microsystems books.

- *Networking on the Sun Workstation: Remote Procedure Call Programming Guide*, (800-1324-03), Sun Microsystems, Inc.
- *Network Programming*, (800-1779-10), Sun Microsystems, Inc.

X Window System Publications

The following list shows X Window System books.

- *Introduction to the X Window System*, Oliver Jones, Prentice-Hall, 1988, ISBN 0-13-499997-5.
- *PEXlib Specification and C Language Binding*, Jeff Stevenson, Hewlett-Packard Company, 1992, SR28-5116.
- *The X Window System Series* (6 volumes), O'Reilly & Associates, 1988, 1989, 1990, ISBN 0-937175-40-4, 0-937175-27-7, 0-937175-28-5, 0-937175-35-6, 0-937175-33-1, 0-937175-35-8.
- *X Protocol Reference Manual*, Adrian Nye, ed. O'Reilly & Associates, Inc., 1990, ISBN 0-937175-50-1.
- *X Window System: C Library and Protocol Reference*, Robert Scheifler, James Gettys, and Ron Newman, DEC Press, 1988, ISBN 1-55558-012-2.
- *X Window System: Programming and Applications with Xt*, Douglas A. Young, Prentice-Hall, 1989, ISBN 0-13-972167-3.
- *X Window System: Programming and Applications with Xt, OSF/Motif Edition*, Douglas A. Young, Prentice-Hall, 1990, ISBN 0-13-497074-8.

- *X Window System Technical Reference*, Steven Mikes, Addison-Wesley, 1990, ISBN 0-201-52370-1.
- *X Window System User's Guide*, Valerie Quercia and Tim O'Reilly, O'Reilly & Associates, Inc., 1990, ISBN 0-937175-14-5.

Network Architecture Publications

The following sections list books associated with network architecture.

Open Systems Interconnection (OSI) Publication

Open Systems Interconnection, Z320-9757.

Systems Network Architecture (SNA) Publications

The following list shows books in the SNA library.

- *Systems Network Architecture: Sessions between Logical Units*, GC20-1868.
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112.
- *Systems Network Architecture Format and Protocol Reference Manual: Management Services*, SC30-3346.
- *Systems Network Architecture Formats*, GA27-3136.
- *Systems Network Architecture Network Product Formats*, LY43-0081.

Communicating Your Comments to IBM

OS/390 TCP/IP OpenEdition
Messages and Codes

Publication No. SC31-8307-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:

  United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**
  - Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

OS/390 TCP/IP OpenEdition
Messages and Codes

Publication No. SC31-8307-00

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.

Help us help you!
SC31-8307-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
Department CGMD
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

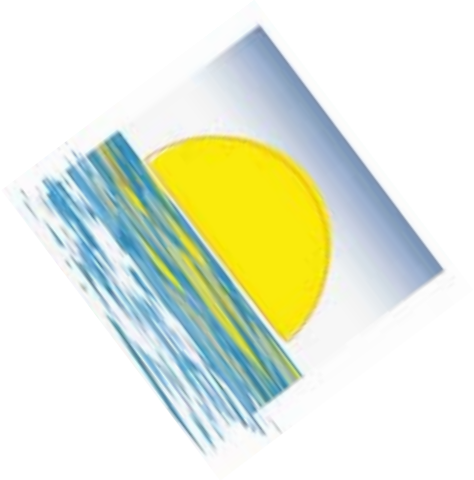
Fold and Tape **Please do not staple** Fold and Tape

SC31-8307-00

Cut or Fold
Along Line

File Number: S390-50
Program Number: 5645-001

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.



SC31-8307-00

Spine information:

OS/390 TCP/IP OpenEdition Messages and Codes